ARAS: FULLY ALGEBRAIC TWO-LEVEL DOMAIN DECOMPOSITION PRECONDITIONING TECHNIQUE WITH APPROXIMATION ON COARSE INTERFACES

THOMAS DUFAUD[1] AND DAMIEN TROMEUR-DERVOUT[1]

[1] UNIVERSITÉ DE LYON, UNIVERSITÉ LYON 1, CNRS, INSTITUT CAMILLE-JORDAN, 43, BOULEVARD DU 11 NOVEMBRE 1918, 69622 VILLEURBANNE, FRANCE

Abstract. This paper focuses on the development of a two-level preconditioner based on a fully algebraical enhancement of a Schwarz domain decomposition method. We consider the purely divergence of a Restricted Additive Scwharz iterative process leading to the preconditioner developped by X.-C. Cai and M. Sarkis in SIAM Journal of Scientific Computing, Vol. 21 no. 2, 1999. The convergence of vectorial sequence of traces of this process on the artificial interface can be accelerated by an Aitken acceleration technique as proposed in the work of M. Garbey and D. Tromeur-Dervout, in International Journal for Numerical Methods in Fluids, Vol. 40, no. 12,2002. We propose a formulation of the Aitken-Schwarz technique as a preconditioning technique called Aitken-RAS [1]. The Aitken acceleration is performed in a reduced space to save computing or permit fully algebraic computation of the accelerated solution without knowledge of the underlying equations. A convergence study of the Aitken-RAS preconditioner is proposed also application on industrial problem.

Key words. Domain decomposition, Restricted Additive Schwarz preconditioner, Aitken-Schwarz method, algebraic multilevel preconditioner;

1. Introduction. The convergence rate of a Krylov method such as GCR and GMRES, developed by Einsenstat & al [18], to solve a linear system $Au = f$, $A = (a_{ij}) \in \mathbb{R}^{m \times m}, u \in \mathbb{R}^m, b \in \mathbb{R}^m$, depends on the matrix eigenvalues distribution. They derived the convergence rate of the GCR and GMRES methods as for the GCR:

$$||r_i||_2 \leq [1 - \frac{\lambda_{min}(M)^2}{\lambda_{min}(M)\lambda_{max}(M) + \rho(R)^2}]^{i/2}||r_0||_2 \leq [1 - \frac{1}{\kappa(M)}]^{i/2}||r_0||_2 \quad (1.1)$$

where $M = (A + A^t)/2$ and $R = (A - A^t)/2$. The GMRES method is mathematically equivalent to the ORTHORES algorithm developed by Young and Cea [40]. Its convergence rate follows the same formula as (1.1) if A is positive real. Otherwise when A is diagonalisable $A = X\Lambda X^{-1}$, the convergence of the GMRES method depends on the distribution of the eigenvalues and the condition number $\kappa(X)$ as follows. Let $\lambda_1, \ldots, \lambda_\mu$ be the eigenvalues of A with a non-positive real part, and let $\Lambda_{\mu 1}, \ldots, \lambda_n$ those with a positive real part belonging to the circle centred in $C > 0$ with a radius R with $C > R$. Then the GMRES convergence rate can be written as:

$$||r_{i1}||_2 \leq \kappa(X) \left[\frac{D}{d}\right]^\mu \left[\frac{R}{C}\right]^{i-\mu} ||r_0||_2 \quad (1.2)$$

with $D = \max_{i=1,\mu; j=\mu 1,n} |\lambda_i - \lambda_j|$ and $d = \min_{i=1,\mu} |\lambda_i|$.(1.1) and (1.2), and, generally speaking, decreases when the condition number $\kappa_2(A) = ||A||_2||A^{-1}||_2$ of the non-singular matrix A increases. This implies the need to reduce the scattering of the eigenvalues distribution in the complex plane in order to improve the convergence rate. This is the goal of a preconditioning technique. The left-preconditioning techniques consist to solve $M^{-1}Au = M^{-1}f$ such that $\kappa_2(M^{-1}A) << \kappa_2(A)$. In this work we focus on the Schwarz

---

[1]This paper extends the proposition of the ARAS preconditioning technique published in T. Dufaud and D. Tromeur-Dervout, Aitken's acceleration of the Resctricted Additive Schwarz preconditioning using coarse approximations on the interface. C. R. Math. Acad. Sci. Paris, Vol. 348, no. 13-14, pages 821-824, 2010, by developing the building of the preconditioner and its theoretical properties. Moreover, it focuses on a fully algebraic technique based on SVD to approximate the solutions. Finally, results on industrial linear systems are provided.

preconditioning techniques and the preconditioning techniques that are related to the Schur complement of the matrix A.

Let us first recall some state of the art about the Schwarz and Aitken-Schwarz solvers, and preconditioners based on the Restricted Additive Schwarz (RAS).

1.1. State of the art of Schwarz and Aitken techniques. First, lets us consider the Generalized Schwarz Alternating Method introduced by Engquist and Zao [19] that gathers several Schwarz techniques (see Quarteroni and Valli [35]). For sake of simplicity, let us consider the case where the whole domain Ω is decomposed into two sub-domains Ω_1 and Ω_2, with overlapping or not, defining two artificial boundaries Γ_1, Γ_2. Let $\Omega_{11} = \Omega_1 \backslash \Omega_2$, $\Omega_{22} = \Omega_2 \backslash \Omega_1$ if there is an overlap. Let $L(x)$ be the continuous operator associated with the discrete operator A. It can be written in its multiplicative version as:

Algorithm 1 GSAM: Multiplicative version

1: DO until convergence
2: Solve

$$L(x)u_1^{2n+1}(x) = f(x),\ \forall x \in \Omega_1, \tag{1.3}$$

$$u_1^{2n+1}(x) = g(x),\ \forall x \in \partial\Omega_1\backslash\Gamma_1, \tag{1.4}$$

$$\Lambda_1 u_1^{2n+1} + \lambda_1 \frac{\partial u_1^{2n+1}(x)}{\partial n_1} = \Lambda_1 u_2^{2n} + \lambda_1 \frac{\partial u_2^{2n}(x)}{\partial n_1},\ \forall x \in \Gamma_1 \tag{1.5}$$

3: Solve

$$L(x)u_2^{2n+2}(x) = f(x),\ \forall x \in \Omega_2, \tag{1.6}$$

$$u_2^{2n+2}(x) = g(x),\ \forall x \in \partial\Omega_2\backslash\Gamma_2, \tag{1.7}$$

$$\Lambda_2 u_2^{2n+2} + \lambda_2 \frac{\partial u_2^{2n+2}(x)}{\partial n_2} = \Lambda_2 u_1^{2n+1} + \lambda_2 \frac{\partial u_1^{2n+1}(x)}{\partial n_2},\ \forall x \in \Gamma_2. \tag{1.8}$$

4: Enddo

---

where Λ_i are some operators and λ_i are constants.

According to the specific choice of the operators Λ_i and the values of scalars λ_i, we obtain the family of Schwarz domain decomposition techniques:

Overlap	Λ_1	Λ_2	λ_1	λ_2	Method
yes	Id	Id	0	0	Schwarz
yes	Id	Id	α	α	ORAS (St Cyr & al [36])
No	Id	0	0	1	Neumann-Dirichlet (Marini & Quarteronni [33])
No	Id	Id	1	1	Modified Schwarz (Lions [31])

TABLE 1.1
Derived methods obtained from the specific choices of the operators Λ_i and the values of scalars λ_i in the GSAM.

If $\Lambda_1 = \Lambda_2 = I$ and $\lambda_1 = \lambda_2 = 0$ then the above multiplicative version is the classical Multiplicative Schwarz. If $\Lambda_1 = \Lambda_2 = constant$ and $\lambda_1 = \lambda_2 = 1$ then it is the modified Schwarz proposed by Lions in [32].

Engquist and Zao [19] showed that with an appropriate choice of the operators Λ_i this

domain decomposition method converges in two iterations. They established the proposition that follows:

PROPOSITION 1.1. *If Λ_1 (or Λ_2) is the Dirichlet to Neumann operator at the artificial boundary Γ_1 (or Γ_2) for the corresponding homogeneous PDE in Ω_2 (or Ω_1) with homogeneous boundary condition on $\partial\Omega_2 \cap \partial\Omega$ (or $\partial\Omega_1 \cap \partial\Omega$) then the Generalized Schwarz Alternating method converges in two steps.*

The GSAM method converges in two steps if the Dirichlet-Neumann operators Λ_i, $i = 1, 2$, are available. These operators are not local to a sub-domain but they link up together all the sub-domains. In practice, some approximations defined algebraically of these operators are used (see Chevalier & Nataf [11], Gander & al [23], Gerardo-Giorda & Nataf [27]).

In the Aitken-Schwarz methodology introduced by Garbey & Tromeur-Dervout [25, 26], only the convergence property of the Schwarz method is used. Consequently, no direct approximation of the Dirichlet-Neumann map is used, but an approximation of the operator of error linked to this Dirichlet-Neumann map is performed. This Aitken-Schwarz methodology is based on the purely linear convergence for the Schwarz Alternating method when the local operators are linear operators.

DEFINITION 1.2. *Let $\left(\left(u_i^k\right)_{i=1,\ldots,n} = u^k\right)_{k\in\mathbb{N}}$ be a vectorial sequence converging toward $(\xi)_{i=1,\ldots,n} = \xi$ purely linearly if*

$$u^{k+1} - \xi = P\left(u^k - \xi\right) \tag{1.9}$$

where $P \in \mathbb{R}^{n \times n}$ is a constant error's transfer operator independent of k and non-singular. We assume that there exists a norm $||.||$ such as $||P|| < 1$. Then P and ξ can be determined from $n + 1$ iterations , using the equations:

$$\left(u^{k+1} - u^k, \ldots, u^2 - u^1\right) = P\left(u^k - u^{k-1}, \ldots, u^1 - u^0\right) \tag{1.10}$$

So, if $\left(u^n - u^{n-1}, \ldots, u^1 - u^0\right)$ is non-singular P can be written as :

$$P = \left(u^{n+1} - u^n, \ldots, u^2 - u^1\right)\left(u^n - u^{n-1}, \ldots, u^1 - u^0\right)^{-1} \tag{1.11}$$

Then if $||P|| < 1$, $(I - P)$ is non singular and it is possible to compute ξ as

$$\xi = (I - P)^{-1}\left(u^{n+1} - Pu^n\right) \tag{1.12}$$

For domain decomposition methods, the vectorial sequences u^n corresponds to the iterated solution at the subdomains artificial interfaces. To apply directly the Aitken's acceleration in the vectorial case, we have to construct the matrix P or an approximation of it, and to apply the Aitken's acceleration (1.12). Algorithm 2 describes the acceleration written in the canonical base of $\mathbb{R}^n$ ("physical space").

Algorithm 2 Vectorial Aitken's acceleration in the physical space

---

Require: $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ an iterative method having a pure linear convergence
Require: $(u^k)_{1\le k \le n+1}$, $n + 1$ successive iterates of $\mathcal{G}$ starting from an arbitrary initial guess u^0
1: Form $E^k = u^{k+1} - u^k$, $0 \le k \le n$
2: **if** $\left[E^{n-1}, \ldots, E^0\right]$ is invertible **then**
3:   $P = \left[E^n, \ldots, E^1\right]\left[E^{n-1}, \ldots, E^0\right]^{-1}$
4:   $u^\infty = (I_n - P)^{-1}(u^{n+1} - Pu^n)$
5: **end if**

The drawback of Algorithm 2 is to be limited to a sequence of small vector size because it needs a number of iterations related to the vector size n. In order to overcome this difficulty, some approximation of the error transfer operator P is proposed using some coarse approximation spaces to represent the solution. Garbey [24] proposed to write the solution in the eigenbasis associated to the part of the separable operator associated to the direction parallel to the artificial interfaces, or with a coarse approximation of the sinus or cosinus expansion of the solution at the interface. Tromeur-Dervout [39] proposed to build an approximation space based on the Singular Value Decomposition (SVD) of the interface solutions of the Schwarz iterates. This last approximation override any constraints about separability of the linear operator A and mesh considerations.

Nevertheless this last techniques fail when:
- the iterative process based on domain decomposition diverges too fast,
- the local solutions are inaccurately solved leading to a less numerically efficient acceleration by the Aitken's process.

In such cases the Aitken-Schwarz method as solver is no longer suitable should be considered as preconditioner of a Krylov method. The purpose of this paper is to detail and to extend the preconditioners based on Schwarz domain decomposition accelerated by Aitken's techniques developed by Dufaud & Tromeur-Dervout [16] with building and approximation of the matrix P arising from the SVD approximation of the Schwarz interface solutions.

1.2. State of the art of preconditioners based on RAS. The techniques of preconditioning that are based on domain decomposition of Schwarz's type have been widely developed this last decade and accelerated multiplicative Schwarz has been "*a consistently good performer*" as said Cai & al [7]. The first type of domain decomposition preconditioning to appear was domain decomposition based on substructuring technique of Bramble & al [5] followed by the Additive Schwarz (AS) preconditioning of Dryja & Widlund [14], Gropp & Keyes [28]. It is built from the adjacency graph $G = (W, E)$ of A, where $W = \{1, 2, ..., m\}$ and $E = \{(i, j) : a_{ij} \neq 0\}$ are the edges and vertices of G. Starting with a non-overlapping partition $W = \cup_{i=1}^{p} W_{i,0}$ and $\delta \geq 0$ given, the overlapping partition $\{W_{i,\delta}\}$ is obtained defining p partitions $W_{i,\delta} \supset W_{i,\delta-1}$ by including all the immediate neighbouring vertices of the vertices in the partition $W_{i,\delta-1}$. Then the restriction operator $R_{i,\delta} : W \to W_{i,\delta}$ defines the local operator $A_{i,\delta} = R_{i,\delta} A R_{i,\delta}^T$, $A_{i,\delta} \in \mathbb{R}^{m_{i,\delta} \times m_{i,\delta}}$ on $W_{i,\delta}$. The AS preconditioning writes: $M_{AS,\delta}^{-1} = \sum_{i=1}^{p} R_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta}$.

Cai & Sarkis [8] introduced the restriction matrix $\tilde{R}_{i,\delta}$ on a non-overlapping sub-domain $W_{i,0}$, and then derived the Restricted Additive Schwarz (RAS) iterative process as:

$$u^k = u^{k-1} + M_{RAS,\delta}^{-1} \left(f - Au^{k-1} \right), \text{ with } M_{RAS,\delta}^{-1} = \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} \qquad (1.13)$$

They showed experimentally that the RAS exhibits a faster convergence than the AS, as Efstathiou & Gander demonstrated in [17] for the Poisson problem, leading to a better preconditioning that depends on the number of sub-domains. We note, that Cai & al [6] develop extensions of RAS for symmetric positive definite problems using the so-called harmonic overlaps (RASHO).

When it is applied to linear problems, the RAS has a pure linear rate of convergence / divergence. When it converges, its convergence can be enhanced with optimized boundary conditions giving the ORAS method of St Cyr & al [36]. In this case, the transmission condition in GSAM takes the form Λ_i to be the normal derivative (Neumann boundary condition).

Then, an optimisation problem is done to minimize the amplification factor of the Schwarz method with this Robin coefficient in the Fourier space. The drawback of this method is that it can only be applied to separable operators, and need regular step size and periodic boundary conditions in the direction orthogonal to the interface to be mathematically valid. Nevertheless, if it is not the case, the parameters in the Robin conditions are set based on this postulate and applied in the current case.

This Neumann-Dirichlet map is related to the Schur complement of the discrete operator (see for example Natarajan [34] or Steinbach [37]).

Saad & Li [30] introduced the SchurRAS method based on the ILU factorisation of the local operators $A_{i,\delta}$ present in the RAS method. Magoules & al [22] introduced the patch substructuring methods and demonstrated its equivalence with the overlapping Schwarz methods. In this work the Dirichlet and Neumann boundary conditions present in the Schwarz alternated method have been replaced by Robin boundary conditions to enhance the convergence rate. The patch method consists in introducing an overlap in the Schur complement technique. These two techniques take care of the data locality in order to avoid global communications involving all sub-domains. Nevertheless, as in the RAS preconditioning technique, the main drawback of this locality is a decreasing of the preconditioning efficiency with respect to the number of sub-domains.

Another related work that takes care to involve all the sub-domains present in the Schur complement is the substructuring method with a suitable preconditioner for the reduced equation of Bramble & al [5], Carvalho & al [9] [10], Khoromskij & Wittum [29]. Let us describe the iterative substructuring method and the preconditioning by the additive Schwarz preconditioner for the Schur complement reduced equation on the interface problem designed by [9]. Let Γ be the set of all the indices of the mesh points which belong to the interfaces between the sub-domains. Grouping together the unknowns associated to points of the mesh corresponding to Γ into the vector u_Γ and the ones corresponding to the other unknowns (corresponding to the points of mesh associated to the interior I of sub-domains) into the vector u_I, we get the reordered problem:

$$\begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_\Gamma \end{pmatrix} \begin{pmatrix} u_I \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_I \\ f_\Gamma \end{pmatrix} \tag{1.14}$$

Eliminating the unknowns u_I from the second block row of (1.14) leads to the following reduced equation for u_Γ:

$$Su_\Gamma = f_\Gamma - A_{\Gamma I} A_{II}^{-1} f_I, \tag{1.15}$$

where

$$S = A_{\Gamma\Gamma} - A_{\Gamma I} A_{II}^{-1} A_{I\Gamma} \tag{1.16}$$

is the Schur complement of the matrix A_{II} in A. Let be $\Gamma_i = \partial\Omega_i \backslash \partial\Omega$. Let $R_{\Gamma_i} : \Gamma \to \Gamma_i$ be the canonical pointwise restriction which maps vectors on Γ into defined vectors on Γ_i, and let be $R_{\Gamma_i}^T : \Gamma_i \to \Gamma$ its transposed. The Schur complement matrix (1.16) can also be written as:

$$S = \sum_{i=1}^p R_{\Gamma_i}^T S^{(i)} R_{\Gamma_i} \tag{1.17}$$

where

$$S^{(i)} = A_{\Gamma_i\Gamma_i}^{(i)} - A_{\Gamma_i i} A_{ii}^{-1} A_{i\Gamma_i} \tag{1.18}$$

5

is referred to the local Schur complement associated with the sub-domain Ω_i. $S^{(i)}$ that involves the submatrices from the local matrix $A^{(i)}$ which is defined as

$$A^{(i)} = \begin{pmatrix} A_{ii} & A_{i\Gamma_i} \\ A_{\Gamma_i i} & A_{\Gamma_i \Gamma_i} \end{pmatrix} \tag{1.19}$$

Then they defined a BPS (Bramble, Pasciak & Schatz [5]) preconditioner which is based on the set V which gathers the cross points between sub-domains (i.e points that belong to more than two sub-domains) and the sets E_i of interface points (without the cross points in V)

$$E_i = (\partial\Omega_j \cap \partial\Omega_l) - V \tag{1.20}$$

$$\Gamma = (\bigcup_{i=1}^{m} E_i) \cup V \tag{1.21}$$

The operator R_i defined the standard pointwise restriction of nodal values on E_i while operator R_V defined the canonical restriction on V. Then a coarse mesh is associated with the sub-domains and an interpolation operator R^T is defined. This operator corresponds to the linear interpolation between two adjacent cross points V_j V_i in order to define values on the edge E_i . This allows to define A_H the Galerkin cross grid operator $A_H = RAR^T$. They deduced a very close variant of BPS preconditioner that can be written as:

$$M_{BPS} = \sum_{E_i} R_i^T S_{ii} R_i + R^T A_H^{-1} R \tag{1.22}$$

They defined a coarse-space operator

$$\Lambda_0 = R_0 S R_0^T \tag{1.23}$$

where $R_0 : U \rightarrow U_0$ is a restriction operator which maps full vector of U into vector in U_0 where U_0 is a q-dimensional subspace of U the algebraical space of nodal vectors where the Schur complement matrix is defined.

$$M_{BPS} = \sum_{E_i} R_i^T \tilde{S}_{ii} R_i + R_0^T \Lambda_0^{-1} R_0 \tag{1.24}$$

where $\tilde{S}_{ii}$ is an approximation of S_{ii}. The definition of U_0 gives different preconditioners: Vertex-based coarse space, sub-domain-based coarse space, edge-based coarse space, depending on the set of points of the interface Γ that are involved. From the implementation practical point of view, the coarse matrix Λ_0 is constructed once and involve matrix vector products of the local Schur complement only.

1. The advantages of this method, is to defined the two-level preconditioner only on the interface Γ. It is intimately related to the Schur complement operator defined on the interface.
2. The drawback is to have to define a priori the coarse space U_0 without any knowledge of the solution behavior. Consequently it can be expensive in term of number of coarse space vectors, specifically for 3D problems where cross-points between sub-domains in 2D, become cross-regions between sub-domains in 3D .

Our approach will follow the same spirit as this two-level preconditioning working only on the interface. But we still work on the system $Ax = b$ and not $Su_\Gamma = g_\Gamma$ and we use an *a posteriori* knowledge of the global Dirichlet to Neumann map that is based on the pure linear

convergence/divergence of the RAS to define the coarse space (equivalent of the definition of U_0).

The plan of this paper is the following. Section 2 will derive the Aitken-Schwarz preconditioning. Section 3 will focus on coarse approximation of the solution at the artificial interfaces, notably with a random set of orthogonal vectors and an orthogonal set of vectors obtained through the SVD of the Schwarz interface solutions. Then, section 4 proposes a study of convergence of the ARAS class preconditioners. Eventually numerical tests are provided on academic problems in section 5 and industrial problems in section 6.

2. Aitken-Schwarz method derived as preconditioning technique. In this section we study the integration of the Aitken's acceleration into a Richardson process in order to formulate a preconditioning technique based on Aitken. More precisely, we propose to enhance the RAS preconditioning technique, presented in section 1, by the Aitken's acceleration. We first present the mechanism of the method and develop the equation to extract a corresponding Richardson's process. Then we point out that the method in its simple form does not exhibit the complete acceleration after one application and need an update as when the method is used as solver. The result is a multiplicative preconditioner based on the Aitken RAS preconditioner. Finally we present those preconditioners in their approximated form in order to save computing.

2.1. The Aitken Restricted Additive Schwarz preconditioner: ARAS. Let $\Gamma_i = (I_{m_{i,\delta}} - R_{i,\delta}^T)W_{i,\delta}$ be the interface associated to $W_{i,\delta}$ and $\Gamma = \cup_{i=1}^p \Gamma_i$ be the global interface. Then $u_{|\Gamma} \in \mathbb{R}^n$ is the restriction of the solution $u \in \mathbb{R}^m$ on the Γ interface and $e_{|\Gamma}^k = u_{|\Gamma}^k - u_{|\Gamma}^\infty$ is the error of an iteration of a RAS iterative process, equation (2.1) at the interface Γ.

$$u^k = u^{k-1} + M_{RAS,\delta}^{-1}(f - Au^{k-1}) \tag{2.1}$$

In section 1 we wrote that the Schwarz iterative method has a pure linear convergence. This property enables us to use the Aitken's technique presented the same section. The previously mentioned $\mathcal{G}$ iterative process is replaced by a RAS iterative process.

Using the linear convergence property of the RAS method, we would like to write a preconditioner which includes the Aitken's acceleration process. We introduce a restriction operator $R_\Gamma \in \mathbb{R}^{n \times m}$ from W to the global artificial interface Γ, with $R_\Gamma R_\Gamma^T = I_n$. The Aitken Restricted Additive Schwarz (ARAS) must generate a sequence of solution on the interface Γ, and accelerate the convergence of the Schwarz process from this original sequence. Then the accelerated solution on the interface replaces the last one. This could be written combining an AS or RAS process eq.(2.2a)) with the Aitken process written in $\mathbb{R}^{m \times m}$ eq.(2.2b) and subtracting the Schwarz solution which is not extrapolated on Γ eq.(2.2c). We can write the following approximation u^* of the solution u:

$$u^* = u^{k-1} + M_{RAS,\delta}^{-1}(f - Au^{k-1}) \tag{2.2a}$$

$$+ R_\Gamma^T (I_n - P)^{-1} \left(u_{|\Gamma}^k - Pu_{|\Gamma}^{k-1} \right) \tag{2.2b}$$

$$- R_\Gamma^T I_n R_\Gamma \left(u^{k-1} + M_{RAS,\delta}^{-1}(f - Au^{k-1}) \right) \tag{2.2c}$$

We would like to write u^* as an iterated solution derived from an iterative process of the form $u^* = u^{k-1} + M_{ARAS,\delta}^{-1}\left(f - Au^{k-1}\right)$, where $M_{ARAS,\delta}^{-1}$ is the Aitken-RAS preconditioner.

First of all, we write an expression of eq.(2.2b) depending on eq.(2.1) and which only involves the iterated solution $u^{k-1} \in \mathbb{R}^m$, as follows:

$$eq.(2.2b) := R_\Gamma^T \left(I_n - P\right)^{-1} \left(u_{|\Gamma}^k - P u_{|\Gamma}^{k-1}\right)$$

$$= R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma \left(R_\Gamma^T I_n R_\Gamma u^k - R_\Gamma^T P R_\Gamma u^{k-1}\right)$$

$$\downarrow \quad \text{with eq.(2.1)}$$

$$= R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma \left(R_\Gamma^T I_n R_\Gamma \left(u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)\right)$$

$$- R_\Gamma^T P R_\Gamma u^{k-1})$$

$$= R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma R_\Gamma^T I_n R_\Gamma \left(u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)$$

$$- R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma R_\Gamma^T P R_\Gamma u^{k-1}$$

$$= R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma \left(u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)$$

$$- R_\Gamma^T \left(I_n - P\right)^{-1} P R_\Gamma u^{k-1}$$

Then, we re-write eq.(2.2) with this new expression of eq.(2.2b) as follows:

$$u^* = u^{k-1} + M_{RAS,\delta}^{-1}(f - Au^{k-1})$$

$$+ R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma \left(u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)$$

$$- R_\Gamma^T \left(I_n - P\right)^{-1} P R_\Gamma u^{k-1} - R_\Gamma^T I_n R_\Gamma \left(u^{k-1} + M_{RAS,\delta}^{-1}(f - Au^{k-1})\right)$$

$$\downarrow \quad \text{factorizing by } \left(u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)$$

$$= \left(I_m - R_\Gamma^T I_n R_\Gamma + R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma\right) \left(u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)$$

$$- R_\Gamma^T \left(I_n - P\right)^{-1} P R_\Gamma u^{k-1}$$

$$\downarrow \quad \text{isolating } u^{k-1} \text{ from } M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

$$= u^{k-1} + \left(-R_\Gamma^T I_n R_\Gamma + R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma - R_\Gamma^T \left(I_n - P\right)^{-1} P R_\Gamma\right) u^{k-1}$$

$$+ \left(I_m - R_\Gamma^T I_n R_\Gamma + R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma\right) M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

One can simplify $E = \left(R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma - R_\Gamma^T \left(I_n - P\right)^{-1} P R_\Gamma\right)$ as follows:

$$E = R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma \left(R_\Gamma^T I_n R_\Gamma - R_\Gamma^T P R_\Gamma\right)$$

$$= R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma R_\Gamma^T \left(I_n - P\right) R_\Gamma$$

$$= R_\Gamma^T \left(I_n - P\right)^{-1} \left(I_n - P\right) R_\Gamma$$

$$= R_\Gamma^T I_n R_\Gamma$$

And then writes,

$$u^* = u^{k-1} + \left(-R_\Gamma^T I_n R_\Gamma + R_\Gamma^T I_n R_\Gamma\right) u^{k-1}$$

$$+ \left(I_m - R_\Gamma^T I_n R_\Gamma + R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma\right) M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

$$= u^{k-1} + \left(I_m - R_\Gamma^T I_n R_\Gamma + R_\Gamma^T \left(I_n - P\right)^{-1} R_\Gamma\right) M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

$$= u^{k-1} + \left(I_m + R_\Gamma^T \left(\left(I_n - P\right)^{-1} - I_n\right) R_\Gamma\right) M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

Hence the formulation eq.(2.2) leads to an expression of an iterated solution u^*:

$$u^* = u^{k-1} + \left(I_m + R_\Gamma^T \left((I_n - P)^{-1} - I_n\right) R_\Gamma\right) M_{RAS,\delta}^{-1} \left(f - Au^{k-1}\right)$$

This iterated solution u^* can be seen as an accelerated solution of the RAS iterative process. Drawing our inspiration from the Stephensen's method [38], we build a new sequence of iterates from the solutions accelerated by the Aitken's acceleration method. Then, one considers u^* as a new u^k and writes the following ARAS iterative process:

$$u^k = u^{k-1} + \left(I_m + R_\Gamma^T \left((I_n - P)^{-1} - I_n\right) R_\Gamma\right) M_{RAS,\delta}^{-1} \left(f - Au^{k-1}\right) \qquad (2.3)$$

Then we defined the ARAS preconditioner as

$$M_{ARAS,\delta}^{-1} = \left(I_m + R_\Gamma^T \left((I_n - P)^{-1} - I_n\right) R_\Gamma\right) \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} \qquad (2.4)$$

Remark 1. The ARAS preconditioner can be considered as a two-level additive preconditioner. The preconditioner consists in computing a solution on an entire domain applying the RAS preconditioner and add components computed only on the interface Γ.

2.2. Composite Multiplicative form of ARAS: ARAS2. If P is known exactly, the ARAS process written in the equation (2.3) needs two steps to converge to the solution u with an initial guess $u^0 = 0$. Then we have:

PROPOSITION 2.1. *If P is known exactly then we have*

$$A^{-1} = \left(2M_{ARAS,\delta}^{-1} - M_{ARAS,\delta}^{-1} A M_{ARAS,\delta}^{-1}\right)$$

that leads $\left(I - M_{ARAS,\delta}^{-1} A\right)$ to be a nilpotent matrix of degree 2.

Proof *We consider the postulate: "If P is known exactly, the ARAS process written in eq.(2.3) needs two steps to converge to the solution with an initial guess $u^0 = 0$".*

We write the two first iterations of the ARAS process for any initial guess $u^0 \in \mathbb{R}^m$ and for all $f \in \mathbb{R}^m$:

$$u^1 = u^0 + M_{ARAS,\delta}^{-1} \left(f - Au^0\right)$$

And the second iterations leads to:

$$\begin{aligned}
u^2 &= u^1 + M_{ARAS,\delta}^{-1} \left(f - Au^1\right) \\
&= u^0 + M_{ARAS,\delta}^{-1} \left(f - Au^0\right) + M_{ARAS,\delta}^{-1} \left(f - A\left(u^0 + M_{ARAS,\delta}^{-1} \left(f - Au^0\right)\right)\right)
\end{aligned}$$

Let $u^0 = 0$, then,

$$\begin{aligned}
u^2 &= M_{ARAS,\delta}^{-1} f + M_{ARAS,\delta}^{-1} \left(f - A\left(M_{ARAS,\delta}^{-1} f\right)\right) \\
&= \left(2M_{ARAS,\delta}^{-1} - M_{ARAS,\delta}^{-1} A M_{ARAS,\delta}^{-1}\right) f \\
&= u
\end{aligned}$$

Since this expression is true for all $f \in \mathbb{R}^m$ we can write:

$$A^{-1} = 2M_{ARAS,\delta}^{-1} - M_{ARAS,\delta}^{-1} A M_{ARAS,\delta}^{-1}$$

Now we can write:

$$u = \left(2M_{ARAS,\delta}^{-1} - M_{ARAS,\delta}^{-1}AM_{ARAS,\delta}^{-1}\right)f$$

$$= \left(I_m + I_m - M_{ARAS,\delta}^{-1}A\right)M_{ARAS,\delta}^{-1}f$$

$$= M_{ARAS,\delta}^{-1}f + \left(I_m - M_{ARAS,\delta}^{-1}A\right)M_{ARAS,\delta}^{-1}f$$

$$\downarrow \text{ with } Au = f$$

$$= M_{ARAS,\delta}^{-1}Au + \left(I_m - M_{ARAS,\delta}^{-1}A\right)M_{ARAS,\delta}^{-1}Au$$

Thus,

$$\left(I_m - M_{ARAS,\delta}^{-1}A\right)u = \left(I_m - M_{ARAS,\delta}^{-1}A\right)M_{ARAS,\delta}^{-1}Au$$

Which is equivalent to

$$0 = \left(I_m - M_{ARAS,\delta}^{-1}A\right)^2 u, \ \forall u \in \mathbb{R}^m$$

Hence $\left(I_m - M_{ARAS,\delta}^{-1}A\right)$ is nilpotent of degree 2. □

The previous proposition leads to an approximation of A^{-1} written from the 2 first iterations of the ARAS iterative process (2.3). Those 2 iterations compute the Schwarz solutions sequence on the interface needed in order to accelerate the Schwarz method by the Aitken's acceleration. We now write 2 iterations of the ARAS iterative process (2.3) for any initial guess and for all $u^{k-1} \in \mathbb{R}^m$.

$$u^{k+1} = u^{k-1} + M_{ARAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

$$\qquad + M_{ARAS,\delta}^{-1}\left(f - A\left(u^{k-1} + M_{ARAS,\delta}^{-1}\left(f - Au^{k-1}\right)\right)\right)$$

$$= u^{k-1} + M_{ARAS,\delta}^{-1}f - M_{ARAS,\delta}^{-1}Au^{k-1}$$

$$\qquad + M_{ARAS,\delta}^{-1}f - M_{ARAS,\delta}^{-1}Au^{k-1} - M_{ARAS,\delta}^{-1}AM_{ARAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

$$= u^{k-1} + 2M_{ARAS,\delta}^{-1}\left(f - Au^{k-1}\right) - M_{ARAS,\delta}^{-1}AM_{ARAS,\delta}^{-1}\left(f - Au^{k-1}\right)$$

$$= u^{k-1} + \left(2M_{ARAS,\delta}^{-1} - M_{ARAS,\delta}^{-1}AM_{ARAS,\delta}^{-1}\right)\left(f - Au^{k-1}\right)$$

Then we defined the ARAS2 preconditioner as

$$M_{ARAS2,\delta}^{-1} = 2M_{ARAS,\delta}^{-1} - M_{ARAS,\delta}^{-1}AM_{ARAS,\delta}^{-1} \qquad (2.5)$$

Remark 2. According to the linear algebra literature about preconditioning technique [4], the ARAS2 preconditioner can be considered as a composite multilevel preconditioner. Actually, the ARAS2 preconditioner is a multiplicative form of ARAS which is itself an additive preconditioner adding an operation on the entire domain with RAS and an operation on a coarse interface with the Aitken formula.

2.3. Approximated form of ARAS and ARAS2. As the previous subsection suggests, since P is known exactly there is no need to use ARAS as a preconditioning technique. Nevertheless, when P is approximated, the Aitken's acceleration of the convergence depends on the local domain solving accuracy, and the cost of the building of an exact P depends on the size n. This is why P is numerically approximated by $P_{\mathbb{U}_q}$ as in [39], defining $q \leq n$ orthogonal vectors $U_i \in \mathbb{R}^n$, defining the columns of the matrix $\mathbb{U}_q \in R^{n \times q}$. Then it makes sense to use $P_{\mathbb{U}_q}$ in the ARAS preconditioning technique to define the ARAS(q) preconditioner:

$$M_{ARAS(q),\delta}^{-1} = \left(I_m + R_\Gamma^T \mathbb{U}_q \left(\left(I_q - P_{\mathbb{U}_q}\right)^{-1} - I_q\right) \mathbb{U}_q^T R_\Gamma\right) \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} \qquad (2.6)$$

and ARAS2(q),

$$M_{ARAS2(q),\delta}^{-1} = 2M_{ARAS(q),\delta}^{-1} - M_{ARAS(q),\delta}^{-1} A M_{ARAS(q),\delta}^{-1} \qquad (2.7)$$

Different kind of approximation techniques of the error transfer operator matrix P have been proposed in the work of [24, 26, 3, 2, 21]. Nevertheless, we are interested in algebraic ways to compute an Aitken acceleration. In section 3 we proposed two fully algebraic approaches.

3. Basis to approximate the interface solution. We focus here on an algebraic way to compute an Aitken acceleration of a sequence of Schwarz solutions on the interface. The global approach consists on an explicit building of $\hat{P}$ computing how the spanning vectors U_i are modified by the Schwarz iteration. Figure 3.1 illustrates the steps for constructing the matrix $\hat{P}$. Step (a) starts from the spanning vector on the interface $R_\Gamma^T U_i$ and gets its value on the interface in the physical space. Then step (b) performs a complete Schwarz iteration with zero local right hand sides and homogeneous boundary conditions on the others artificial interfaces. Step (c) decomposes the trace solution on the interface in the spanning vector set $\mathbb{U}_q$. Thus, we obtain the column i of the matrix $\hat{P}$.
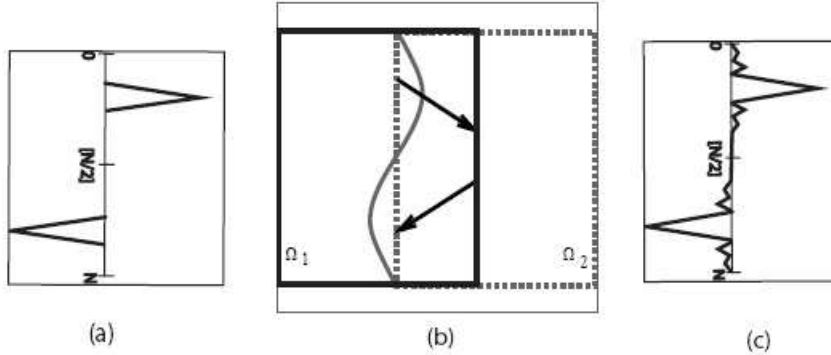

FIGURE 3.1. *Steps to build the $\hat{P}$ matrix*

The full computation of $\hat{P}$ can be done in parallel, but it needs as much local domain solution as the number of interface points (i.e the size of the matrix $\hat{P}$).

Its adaptive computation is required to save computing. This methodology was first used with Fourier basis functions [21, 15]. This section focuses on the definition of orthogonal "base" $\mathbb{U}_q$ that will extend this adaptive computation in a general context. In the following,

we use the term "base" to denote a spanning vectors set that defines the approximation space. The key point of these preconditioners's efficiency is the choice of this orthogonal "base" $\mathbb{U}_q$. It must be sufficiently rich to numerically represent the solution at the interface, but it has to be not too large for the computation's efficiency.

We first propose a "naive" approach consisting of selecting an arbitrary set of orthogonalized random vectors to generate the space to approximate the solution. Secondly, we represent the solution in a space arising from the singular value decomposition of a sequence of Schwarz solutions. Doing this, we assume to represent the main modes of the solutions.

3.1. Orthogonal "base" arising from an arbitrary coarse algebraic approximation of the interface.. A choice consists in having a coarse representation of the interface's solution $u \in \mathbb{R}^n$ from an algebraical point of view. Nevertheless, it is not possible to take a subset of q vectors of the canonical base of $\mathbb{R}^n$, as if some components of u are not reachable by the "base" $\mathbb{U}_q$, then the approximation $||u - \mathbb{U}_q(\mathbb{U}_q^t u)||$ will be very bad. This reason leads us to define $\mathbb{U}_q$ as a set of orthogonal vectors where each component is coming from a random process in order that each vector can contribute to a part of the searched solution at the interface. We split q such as $q = \sum_{i=1}^{p} q_i$ and we associate q_i random vectors to the interface Γ_i, $1 \leq i \leq p$. Then these q_i vectors are orthogonalized to form q_i columns of the "base" $\mathbb{U}_q$.

This strategy is hazardous but can be a simple way to improve the convergence of a Schwarz process without knowledge of the problem and the mesh.

The orthogonal "base" $\mathbb{U}_q$ is obtained applying the same principle as illustrated in Figure 3.1, leading to Algorithm 3.

Algorithm 3 Vectorial Aitken's acceleration in an arbitrary built space without inversion

---

Require: $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ an iterative method having a pure linear convergence
 1: Compute q random vectors $v_i \in \mathbb{R}^n$ following a uniform law on $[0, 1]$
 2: Orthogonalize those q vectors to form $\mathbb{U}_q \in \mathbb{R}^{n \times q}$
 3: Apply **one** iterate of $\mathcal{G}$ on homogeneous problem, $\mathbb{U}_q \to W = \mathcal{G}(\mathbb{U}_q)$
 4: Set $\hat{P} = \mathbb{U}_q^t W$
 5: $\hat{\xi} = (I_q - \hat{P})^{-1} (\hat{u}^1 - \hat{P} \hat{u}^0)$ {Aitken Formula}
 6: $\xi = \mathbb{U}_q \hat{\xi}$

The lack of this method is that there is no possibility to control the quality of the base to perform the acceleration. A more controllable method will be preferred. In the following subsection, we propose a different starting point to build the base. The main idea will be in the fact that we can compress the vectorial sequence using a Singular Value Decomposition. Since the $\mathbb{U}_q$ matrix is built, P is built the same way.

3.2. Approximation compressing the vectorial sequence. A totally algebraic method based on the Singular Value Decomposition of the Schwarz solutions on the interface has been proposed when the modes of the error could be strongly coupled [39]. This method offers the possibility for the Aitken Schwarz method to be used on a large class of problem without mesh consideration. Moreover, when computing an Aitken acceleration, the main difficulty is to invert the matrix $[E^{n-1}, \ldots, E^0]$ which can be close to singular. In a computation, most of the time is consumed solving some noise that does not actually contribute to the solution. The singular value decomposition offer a tools to concentrate the effort only on the main parts of the solution.

3.2.1. The singular value decomposition. A singular-value decomposition (SVD) of a real $n \times m$ $(n > m)$ matrix A is its factorization into the product of three matrices:

$$A = \mathbb{U}\Sigma\mathbb{V}^*, \tag{3.1}$$

where $\mathbb{U} = [U_1, \ldots, U_m]$ is a $n \times m$ matrix with orthonormal columns, Σ is a $n \times m$ non-negative diagonal matrix with $\Sigma_{ii} = \sigma_i$, $1 \le i \le m$ and the $m \times m$ matrix $\mathbb{V} = [V_1, \ldots, V_m]$ is orthogonal. The left $\mathbb{U}$ and right $\mathbb{V}$ singular vectors are the eigenvectors of AA^* and A^*A respectively. It readily follows that $Av_i = \sigma_i u_i$, $1 \le i \le m$

We are going to recall some properties of the SVD. Assume that the $\sigma_i, 1 \le i \le m$ are ordered in decreasing order and there exists r such that $\sigma_r > 0$ while $\sigma_r + 1 = 0$. Then A can be decomposed in a dyadic decomposition:

$$A = \sigma_1 U_1 V_1^* + \sigma_2 U_2 V_2^* + \ldots + \sigma_r U_r V_r^*. \tag{3.2}$$

This means that SVD provides a way to find optimal lower dimensional approximations of a given series of data. More precisely, it produces an orthonormal base for representing the data series in a certain least squares optimal sense. This can be summarized by the theorem of Schmidt-Eckart-Young-Mirsky:

THEOREM 3.1. *A non unique minimizer X_* of the problem $\min_{X, rank X = k} ||A - X||_2 = \sigma_{k+1}(A)$, provided that $\sigma_k > \sigma_{k+1}$, is obtained by truncating the dyadic decomposition of (3.2) to contain the first k terms: $X_* = \sigma_1 U_1 V_1^* + \sigma_2 U_2 V_2^* + \ldots + \sigma_k U_k V_k^*$* The SVD of a matrix is well conditioned with respect to perturbations of its entries. Consider the matrix $A, B \in \mathbb{R}^n$, the Fan inequalities write $\sigma_{r+s+1}(A + B) \le \sigma_{r+1}(A) + \sigma_{s+1}(B)$ with $r, s \ge 0$, $r + s + 1 \le n$. Considering the perturbation matrix E such that $||E|| = O(\epsilon)$, then $|\sigma_i(A + E) - \sigma_i(A)| \le \sigma_1(E) = ||E||_2, \forall i$. This property does not hold for eigenvalues decomposition where small perturbations in the matrix entries can cause a large change in the eigenvalues.

This property allows us to search the acceleration of the convergence of the sequence of vectors in the base linked to its SVD.

PROPOSITION 3.2. *Let $(u^k)_{1 \le k \le q}$ q successive iterates satisfying the pure linear convergence property: $u^k - u^\infty = P(u^{k-1} - u^\infty)$. Then there exists an orthogonal base $\mathbb{U}_q = [U^1, U^2, \ldots, U^q]$ of a subset of $\mathbb{R}^n$ such that*
- *$\alpha_l^k = \sigma_l V_{kl}^*$ with $(\sigma_l)_{l \in \mathbb{N}}$ decreasing and $|V_{kl}^*| \le 1 \Rightarrow \forall l \in \{1, ..., q\}$, $|\alpha_l^k| \le |\sigma_l|$*
- *$u^k = \sum_{l=1}^q \alpha_l^k U^l, \forall k \in \{1, ..., q\}$*

One can write:

$$(\alpha_1^{k+1} - \alpha_1^k, \ldots, \alpha_q^{k+1} - \alpha_q^k)^T = \hat{P}(\alpha_1^k - \alpha_1^{k-1}, \ldots, \alpha_q^k - \alpha_q^{k-1})^T \tag{3.3}$$

where $\hat{P} \stackrel{def}{=} \mathbb{U}_q^* P \mathbb{U}_q$. Moreover $(\alpha_1^\infty, \ldots, \alpha_q^\infty)^T$ obtained by the acceleration process represents the projection of the limit of the sequence of vectors in the space generated by $\mathbb{U}_q$.

Proof *By theorem 3.1.3 there exists a SVD decomposition of $[u^1, \ldots, u^q] = \mathbb{U}_q \Sigma \mathbb{V}^*$ and we can identify α_l^k as $\sigma_l V_{kl}^*$. The orthonormal property of $\mathbb{V}$ associated to the decrease of σ_l with increasing l leads to have α_l^k bounded by $|\sigma_l|$: $\forall l \in \{1, ..., q\}$, $|\alpha_l^k| \le |\sigma_l|$.*
Taking the pure linear convergence of u^k in the matrix form, and applying $\mathbb{U}_q$ leads to:

$$\mathbb{U}_q^*(u^k - u^\infty) = \mathbb{U}_q^* P \mathbb{U}_q \mathbb{U}_q^*(u^{k-1} - u^\infty) \tag{3.4}$$

$$(\alpha_1^k - \gamma_1^\infty, \ldots, \alpha_q^k - \gamma_q^\infty)^T = \hat{P}(\alpha_1^{k-1} - \gamma_1^\infty, \ldots, \alpha_q^{k-1} - \gamma_q^\infty)^T \tag{3.5}$$

where $(\gamma_j^\infty)_{1 \le j \le q}$ represents the projection of u^∞ on the span $\{U_1, \dots, U_q\}$. []

We can then derive Algorithm 4. This algorithm is similar to Algorithm 2 since the error transfer operator is defined using the errors of the linear iterative process in a space arising from the Singular Values Decomposition of $q + 2$ successive iterates. Therefore the third step of Algorithm 2 is equivalent to the sixth step of Algorithm 4.

Algorithm 4 Vectorial Aitken's acceleration in the SVD space with inversion

---

Require: $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ an iterative method having a pure linear convergence
Require: $(u^k)_{1 \le k \le q+2}$, $q + 2$ successive iterates of $\mathcal{G}$ to solve the linear system $Au = f$, starting from an arbitrary initial guess u^0

1: Form the SVD decomposition of $Y = [u^{q+2}, \dots, u^1] = \mathbb{U}SV^T$
2: Set l the index such that $l = \max_{1 \le i \le m+1} \{S(i,i) > tol\}$, $\{$ex.:$tol = 10^{-12}$.$\}$
3: Set $\hat{Y}_{1:l,1:l+2} = S_{1:l,1:l} V^t_{1:l,q-l:q+2}$
4: Set $\hat{E}_{1:l,1:l+1} = \hat{Y}_{1:l,2:l+2} - \hat{Y}_{1:l,1:l+1}$
5: **if** $\hat{E}_{1:l,1:l}$ is non singular **then**
6: $\quad \hat{P} = \hat{E}_{1:l,2:l+1} \hat{E}^{-1}_{1:l,1:l}$
7: $\quad \hat{y}^\infty_{1:l,1} = (I_l - \hat{P})^{-1} (\hat{Y}_{1:l,l+1} - \hat{P}\hat{Y}_{1:l,l})$ {Aitken Formula}
8: $\quad u^\infty = \mathbb{U}_{:,1:l} \, \hat{y}^\infty_{1:l,1}$
9: **end if**

PROPOSITION 3.3. *Successive applications of Algorithm 4 converge to the limit u^∞.*
Proof *As the sequence of vector u^k converges to a limit u^∞ then we can write*

$$\Xi = [u^1, \dots, u^q] = [u^\infty, \dots, u^\infty] + E$$

where E is a $n \times q$ matrix with decreasing coefficients with respect to the columns. The SVD of $\Xi^\infty = [u^\infty, \dots, u^\infty]$ leads to have $U^1 = u^\infty$ and $\sigma_i(\Xi^\infty) = 0$, $i \ge 2$. The fan inequalities lead to have $\sigma_i(\Xi) \le \sigma_1(E) = ||E||_2, i \ge 2$. Consequently, successive applications of Algorithm 4 decrease the number of non zero singular values at each application. []

In Algorithm 4 the building of P needs the inversion of the matrix $\hat{E}_{1:l,1:l}$ which can contain very small singular values even if we selected those greater than a certain tolerance. This singular value can deteriorate the ability of P to accelerate the convergence. If it is the case, we can proceed inverting this matrix with its SVD, replacing by zeros the singular values less than a tolerance instead of inverting them (see numerical recipes [20]). A more robust algorithm can be obtained without inverting $\hat{E}_{1:l,1:l}$. It consists in building P by applying the iterative method $\mathcal{G}$ to the selected columns of $\mathbb{U}_q$ that appears in Algorithm 4. Then $\hat{P} = \mathbb{U}^*_{1:n,1:l} \mathcal{G}(\mathbb{U}_{1:n,1:l})$ as done in Algorithm 5.

4. Convergence of ARAS and ARAS2 and their approximated form. As an enhancement of the RAS preconditioning technique, ARAS and ARAS2 should have a better convergence rate than the RAS technique. We formulate the convergence rate of a RAS technique considering the linear convergence of the Restricted Additive Schwarz method and extend this formulation to the Aitken's technique. Then we propose a relation between the spectral radius of those methods.

In the following we note

$$T_* = (I - M_*^{-1}A) \tag{4.1}$$

Any Richardson's process can be written as

$$u^k = T_* u^{k-1} + c, \text{ where } c \in \mathbb{R}^n \text{is constant} \tag{4.2}$$

Algorithm 5 Vectorial Aitken acceleration in the SVD space without inversion

---

Require: $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ an iterative method having a pure linear convergence

Require: $(u^k)_{1 \leq k \leq q+2}$, $q + 2$ successive iterates of $\mathcal{G}$ to solve the linear system $Au = f$, starting from an arbitrary initial guess u^0

1: Form the SVD decomposition of $Y = \left[u^{q+2}, \dots, u^1\right] = \mathbb{U}SV^T$
2: Set the index l such that $l = \max_{1 \leq i \leq q+1} \{S(i,i) > tol\}$, {ex.:$tol = 10^{-12}$.}
3: Apply **one** iterate of $\mathcal{G}$ on homogeneous problem, with $l + 2$ initial guesses $\mathbb{U}_{:,1:l} \to W_{:,1:l} = \mathcal{G}(\mathbb{U}_{:,1:l})$
4: Set $\hat{P} = \mathbb{U}_{:,1:l}^t W_{:,1:l}$
5: Set $\hat{Y}_{1:l,1:2} = S_{1:l,1:l}V_{1:l,q+1:q+2}^t$
6: $\hat{y}_{1:l,1}^{\infty} = (I_l - \hat{P})^{-1} \, (\hat{Y}_{1:l,2} - \hat{P} \, \hat{Y}_{1:l,1})$ {Aitken Formula}
7: $u^{\infty} = \mathbb{U}_{:,1:l} \, \hat{y}_{1:l,1}^{\infty}$

Remark 3. As the ARAS2 iterative process correspond to 2 *iterations of the ARAS process, we notice that* $T_{ARAS2} = T_{ARAS}^2$.

4.1. Ideal case. When building the ARAS preconditioner, we exhibit the fact that T_{ARAS}■ is nilpotent when the error's transfer operator on the interface Γ, if P is considered exact. This property gives the following proposition:

PROPOSITION 4.1. *If P is known exactly then,*

$$\rho(T_{ARAS2}) = \rho(T_{ARAS}) = 0 \tag{4.3}$$

Proof *If P is known exactly then Proposition 2.1 is verified and T_{ARAS} and T_{ARAS2} are nilpotent. The spectral radius of a nilpotent matrix is equal to 0.* []

Remark 4. Obviously, $\rho(T_{ARAS2}) = \rho(T_{ARAS}) < \rho(T_{RAS})$

But the matrix P is often numerically computed and then $\rho(T_{ARAS})$ is no longer equal to 0. The value of $\rho(T_{ARAS})$ depends on the accuracy of the local domain solutions and when P is written in another space, depends on the quality of this space. In the following we propose a framework to study the convergence of $T_{ARAS(q)}$ and $T_{ARAS2(q)}$. The goal is to provide key elements to understand the influence of approximating P in an orthogonal base on the preconditioner.

4.2. Convergence of RAS for an elliptic operator. In this subsection we express the convergence rate of a RAS iterative process considering its convergence on the artificial interfaces in proposition 4.2. Since we can link the convergence of RAS on the entire domain to the convergence on the interface, it becomes possible to study the effect of modifying the error's transfer operator P.

PROPOSITION 4.2. *Let A be a discretized operator of an elliptic problem on a domain Ω. Let us consider a RAS iterative process such as $T_{RAS} = I - M_{RAS}^{-1}A$ defined on p domains. The data dependencies between domains is located on an artificial interface Γ. Then there exists an error's transfer operator on the interface Γ, P such as there exists a norm $||.||$ for which $||P|| < 1$. The convergence rate of T_{RAS} is*

$$\rho(T_{RAS}) = \max\{|\lambda| : \lambda \in \lambda(P)\} \tag{4.4}$$

Proof *In the case of elliptic problem the maximum principle is observed. Then, for the Schwarz method the error is maximal on artificial interfaces. We write the error of a Schwarz process starting for the definition of the RAS iterative method:*

$$u^{k+1} = T_{RAS}u^k + M_{RAS}^{-1}b \tag{4.5}$$

The convergence of such a process is given by [1] [12]:

$$e^k = T_{RAS}^k e^0 \tag{4.6}$$

On the interface, one can write:

$$e_{|\Gamma}^k = P^k e_{|\Gamma}^0 \tag{4.7}$$

The error is maximal on the interface thus,

$$||e^k||_\infty = ||e_{|\Gamma}^k||_\infty \tag{4.8}$$

Equations (4.6), (4.7) and (4.8) lead to

$$||T_{RAS}^k e^0||_\infty = ||P^k e_{|\Gamma}^0||_\infty < ||P^k||_\infty ||e_{|\Gamma}^0||_\infty \tag{4.9}$$

Then we can write,

$$\sup_{||e^0||_\infty = 1} \left(||T_{RAS}^k e^0||_\infty \right) = \sup_{||e_{|\Gamma}^0||_\infty = 1} \left(||P^k e_{|\Gamma}^0||_\infty \right) \tag{4.10}$$

$$= ||P^k||_\infty \tag{4.11}$$

Hence,

$$\lim_{k->\infty} ||P^k||_\infty^{\frac{1}{k}} = \rho(P) = \rho(T_{RAS}) \tag{4.12}$$

□

4.3. Convergence of ARAS and ARAS2 in their approximated form. As we mentioned previously, there exist different approaches to approximate the error transfer operator P. For a fully algebraic approach, one will choose the approximation of the operator in a basis built explicitly as we described in 3. When it is possible, one can build a complete base analytically and make an approximation of the operator in this base by truncation. In the following, we choose the analytical approach to study the convergence of the method.

Here we focus on elliptic and separable operators. We propose a theorem giving the convergence rate of an ARAS iterative process when the error's transfer operator can be exactly computed in a space spanned by the eigenvectors of P and then truncated to provide an approximation of the error transfer operator in the physical space.

THEOREM 4.3. *Let A be a discretized operator of an elliptic problem on a domain Ω. Let us consider a RAS iterative process such as $T_{RAS} = I - M_{RAS}^{-1}A$ defined on p domains. Let the error transfer operator P on an interface Γ be diagonalisable. If P is diagonalisable, its decomposition in eigenvalues leads to have $P = \mathbb{U}\hat{P}\mathbb{U}^{-1}$ where for $i \in [\![1,n]\!]$, $\hat{P} = diag(\lambda_i)$. The error on the interface Γ in the approximation space follows $\hat{\hat{e}}_{|\Gamma}^{k+1} = \hat{P}\hat{\hat{e}}_{|\Gamma}^k$. Each mode converges linearly and independently from the others following $\hat{\hat{e}}_{|\Gamma,i}^{k+1} = \lambda_i \hat{\hat{e}}_{|\Gamma,i}^k$. Let $Q_\lambda \in \mathbb{R}^{n \times n}$ be a diagonal matrix such that $q_l = 1$ if $1 \le l \le q$ and $q_l = 0$ if $q < l$. And let $\bar{Q}_\lambda = I_n - Q_\lambda$. A coarse approximation of $\hat{P}$ can be done choosing a set of q strong modes as $\tilde{P} = Q_\lambda \hat{P}$. Writing the preconditioner as:*

$$M_{ARAS(q),\delta}^{-1} = \left(I_m + R_\Gamma^T \mathbb{U} \left(\left(I_n - \tilde{P} \right)^{-1} - I_n \right) \mathbb{U}^{-1} R_\Gamma \right) \sum_{i=1}^p \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} \tag{4.13}$$

The spectral radius of $T_{ARAS(q)}$ is :

$$\rho(T_{ARAS(q)}) = \rho(\bar{Q}_\lambda \hat{\tilde{P}}) = \lambda_{q+1} < \min\{|\lambda| : \lambda \in \lambda(Q_\lambda \hat{\tilde{P}})\} \qquad (4.14)$$

Proof *We consider the assumptions of the theorem and the formula of the approximated pre-conditioner 4.13. The equation* (2.2) *can be written for ARAS(q) such as:*

$$\begin{aligned}
u^* &= T_{RAS}u^{k-1} + M_{RAS}^{-1}b \\
&+ R_\Gamma^T (I_n - P)^{-1} \left(u_{|\Gamma}^k - Pu_{|\Gamma}^{k-1} \right) R_\Gamma \\
&- R_\Gamma^T I_n R_\Gamma \left(T_{RAS}u^{k-1} + M_{RAS}^{-1}b \right)
\end{aligned}$$

We consider that on the interface:

$$\left(T_{RAS}u^{k-1} + M_{RAS}^{-1}b \right)_{|\Gamma} = Pu_{|\Gamma}^{k-1} + c \qquad (4.15)$$

With $c \in \mathbb{R}^n$, a constant vector independent of $u_{|\Gamma}$.

Extracting the interface's solution in the approximation space,

$$\hat{u}_{|\Gamma}^* = \hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} + \hat{\tilde{c}} + \left(I_n - \tilde{P} \right)^{-1} \left(\hat{u}_{|\Gamma}^k - \tilde{P}\hat{u}_{|\Gamma}^{k-1} \right) - Q_\lambda \hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} - Q_\lambda \hat{\tilde{c}}$$

Then,

$$\hat{u}_{|\Gamma}^k = \bar{Q}_\lambda \hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} + Q_\lambda \hat{u}_{|\Gamma}^\infty + \bar{Q}_\lambda \hat{\tilde{c}} \qquad (4.16)$$

The error on the interface is

$$\hat{u}_{|\Gamma}^\infty - \hat{u}_{|\Gamma}^k = \hat{u}_{|\Gamma}^\infty - \bar{Q}_\lambda \hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} - Q_\lambda \hat{u}_{|\Gamma}^\infty - \bar{Q}_\lambda \hat{\tilde{c}}$$

Thus,

$$\hat{e}_{|\Gamma}^k = \bar{Q}_\lambda \left(\hat{u}_{|\Gamma}^\infty - \hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} - \hat{\tilde{c}} \right) \qquad (4.17)$$

Regarding equation (4.15), $\hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} + \hat{\tilde{c}} = \hat{u}_{|\Gamma}^k$, and then,

$$\hat{u}_{|\Gamma}^\infty - \hat{\tilde{P}}\hat{u}_{|\Gamma}^{k-1} - \hat{\tilde{c}} = \hat{e}_\Gamma^k = \hat{\tilde{P}}\hat{e}_\Gamma^k \qquad (4.18)$$

Hence we write,

$$\hat{e}_{|\Gamma}^k = \bar{Q}_\lambda \hat{\tilde{P}}\hat{e}_{|\Gamma}^{k-1} \qquad (4.19)$$

We showed that the ARAS iterative process has an error's transfer operator, $\bar{Q}_\lambda \hat{\tilde{P}}$, equal to the part of the error's transfer operator $\hat{\tilde{P}}$ that we did not compute. We note that $||\bar{Q}_\lambda \hat{\tilde{P}}|| \leq ||\hat{\tilde{P}}|| < 1$.

A is a discretized operator of an elliptic problem, then we can apply Proposition 4.2 and write:

$$\rho(T_{ARAS(q)}) = \rho(\bar{Q}_\lambda \hat{\tilde{P}}) \qquad (4.20)$$

Then we proof equation 4.14. []

Remark 5. For a separable operator, the error transfer operator on an interface between two domains is diagonalisable [24]. Then, the error transfer operator P on an interface Γ is diagonalisable for a global operator which is separable and for which the interfaces of all the domains are parallel to one discretization direction.

4.4. Convergence study in the case of the $2D$ **Poisson's equation.** We consider a simple case of an elliptic problem on a rectangle defined in equation (4.21). The problem is discretized by $2D$ finite differences and decomposed into 2 domains Ω_1 and Ω_2 of the same size.

$$\begin{cases} -\triangle u &= f, \text{ in } \Omega = [0,1] \times [0,\pi] \\ u &= 0, \text{ on } \partial\Omega \end{cases} \tag{4.21}$$
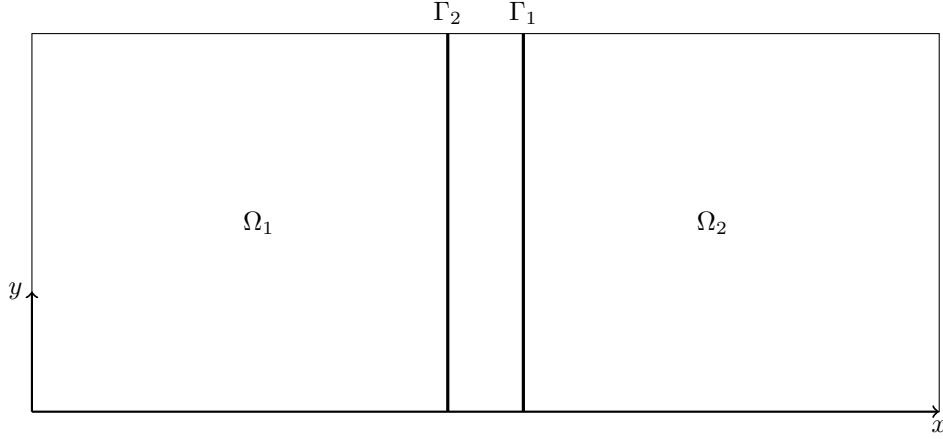

FIGURE 4.1. *2D domain decomposition for Poisson's equation*

In the Fourier base, for a separable operator with 2 artificial interfaces, the acceleration is written with $\hat{P} = \begin{pmatrix} 0 & \hat{P}_{\Gamma_2} \\ \hat{P}_{\Gamma_1} & 0 \end{pmatrix}$ such as

$$\begin{pmatrix} \hat{e}^k_{\Gamma_1} \\ \hat{e}^k_{\Gamma_2} \end{pmatrix} = \begin{pmatrix} 0 & \hat{P}_{\Gamma_2} \\ \hat{P}_{\Gamma_1} & 0 \end{pmatrix} \begin{pmatrix} \hat{e}^{k-1}_{\Gamma_1} \\ \hat{e}^{k-1}_{\Gamma_2} \end{pmatrix} \tag{4.22}$$

We study the case of a regular grid and write the semi-discretized $2D$ Poisson's operator on $[0, \Gamma_1] \times [0, \pi] \cup [\Gamma_2, 1] \times [0, \pi]$, $\Gamma_1 > \Gamma_2$. We consider $N_x + 1$ in the direction x. The overlap in terms of number of step size in direction x is denoted by δ. And we denote by λ_l the eigenvalues of the discretized operator $-\frac{\partial^2}{\partial y^2}$ considering a second order finite difference scheme. We find the coefficient of the matrices $\hat{P}_{\Gamma_i}$ solving the equations:

$$\begin{cases} -\frac{\hat{u}^1_{i+1,l} - 2\hat{u}^1_{i,l} + \hat{u}^1_{i-1,l}}{h_x^2} + \lambda_l \hat{u}^1_{i,l} &= 0 \text{ , } i \in [\![1, N_x - 1]\!] \\ \hat{u}^1_{0,l} &= 0 \\ \hat{u}^1_{N_x,l} &= 1 \end{cases} \tag{4.23}$$

$$\begin{cases} -\frac{\hat{u}^2_{i+1,l} - 2\hat{u}^2_{i,l} + \hat{u}^2_{i-1,l}}{h_x^2} + \lambda_l \hat{u}^2_{i,l} &= 0 \text{ , } i \in [\![1, N_x - 1]\!] \\ \hat{u}^2_{0,l} &= 1 \\ \hat{u}^2_{N_x,l} &= 0 \end{cases} \tag{4.24}$$

The roots of those equations are such as,

$$r_1 = \frac{2 + \lambda_l h_x^2 + \sqrt{\lambda_l^2 h_x^4 + 4\lambda_l h_x^2}}{2} \tag{4.25}$$

$$r_2 = \frac{2 + \lambda_l h_x^2 - \sqrt{\lambda_l^2 h_x^4 + 4\lambda_l h_x^2}}{2} \tag{4.26}$$

Then for the first domain Ω_1 the solutions have the form:

$$\hat{u}_{l,j}^1 = \frac{r_1^j - r_2^j}{r_1^{N_x} - r_2^{N_x}} \tag{4.27}$$

Thus,

$$\hat{u}_{N_x-\delta,l}^1 = \frac{r_1^{N_x-\delta} - r_2^{N_x-\delta}}{r_1^{N_x} - r_2^{N_x}} \tag{4.28}$$

$\hat{P}_{\Gamma_1}$ appears to be diagonal and its diagonal coefficients δ_{l,Γ_1} can be analytically derived such as:

$$\delta_{l,\Gamma_1} = \hat{u}_{N_x-\delta,l}^1 * \hat{u}_{\delta,l}^2 \tag{4.29}$$

$$= \left(\frac{r_1^{N_x-\delta} - r_2^{N_x-\delta}}{r_1^{N_x} - r_2^{N_x}} \right) \left(\frac{-r_2^{N_x} r_1^\delta + r_1^{N_x} r_2^\delta}{r_1^{N_x} - r_2^{N_x}} \right) \tag{4.30}$$

Exactly the same development can be done to find δ_{l,Γ_2}.

Remark 6. If the two domains have the same size, then $\delta_{l,\Gamma_1} = \delta_{l,\Gamma_2}$.

The $\hat{P}$ matrix has the form:

$$\hat{P} = \begin{pmatrix} 0 & & & \delta_{1,\Gamma_2} & & \\ & \ddots & & & \ddots & \\ & & 0 & & & \delta_{n,\Gamma_2} \\ \delta_{1,\Gamma_1} & & & 0 & & \\ & \ddots & & & \ddots & \\ & & \delta_{n,\Gamma_1} & & & 0 \end{pmatrix} \text{, with } 1 > \delta_{1,\Gamma_i} \geq ... \geq \delta_{n,\Gamma_i}. \tag{4.31}$$

For the sake of simplicity we consider that the domain Ω_1 and Ω_2 have the same size. We note $\delta_l = \delta_{l,\Gamma_1} = \delta_{l,\Gamma_2}$. Then we can calculate the determinant of $(\hat{P} - \lambda I_n)$:

$$det(\hat{P} - \lambda I_n) = \prod_{l=1}^{n} (\delta_l - \lambda)(\delta_l + \lambda)$$

Hence the spectrum is $\{\delta_1, ..., \delta_n, -\delta_1, ..., -\delta_n\}$.

We showed that the error's transfer operator is diagonalisable for this problem. P can be written in a base $\mathbb{U}$ of eigenvectors as follows:

$$P = \mathbb{U}\hat{P}\mathbb{U}^{-1}, \text{ with } \hat{P} = diag(\delta_1, ..., \delta_n, -\delta_1, ..., -\delta_n) \tag{4.32}$$

Then we can estimate the convergence rate of the RAS, ARAS(q) and ARAS2(q) applying Theorem 4.3.

$$\rho(T_{RAS}) = \delta_1 \tag{4.33}$$

$$\rho(T_{ARAS(q)}) = \delta_{q+1} \tag{4.34}$$

$$\rho(T_{ARAS2(q)}) = \delta_{q+1}^2 \tag{4.35}$$

Because the eigenvalues and the values of $\hat{P}_{\Gamma_i}$ are equal we can verify a correspondence between the approximation by truncation in the eigenvectors space and the Fourier space.

19

Selecting the first Fourier mode corresponds to selecting the highest eigenvalues. Let us introduce the transfer matrix C_{Γ_i} from the real space to the Fourier space and the transfer matrix D_{Γ_i} from the Fourier space to the real space:

$$
C_{\Gamma_i} \quad : \quad
\begin{array}{c}
\mathbb{R}^n \longrightarrow \mathbb{C}^n \\
e_{|\Gamma_i} \longmapsto \hat{e}_{|\Gamma_i}
\end{array}
\quad \text{and} \quad
D_{\Gamma_i} \quad : \quad
\begin{array}{c}
\mathbb{C}^n \longrightarrow \mathbb{R}^n \\
\hat{e}_{|\Gamma_i} \longmapsto e_{|\Gamma_i}
\end{array}
$$

Then we write

$$
P = D\hat{P}C = \begin{pmatrix} D_{\Gamma_1} & 0 \\ 0 & D_{\Gamma_2} \end{pmatrix} \begin{pmatrix} 0 & \hat{P}_{\Gamma_2} \\ \hat{P}_{\Gamma_1} & 0 \end{pmatrix} \begin{pmatrix} C_{\Gamma_1} & 0 \\ 0 & C_{\Gamma_2} \end{pmatrix} \tag{4.36}
$$

The approximation is done by applying the operator

$$
Q_{\mathcal{F}} = \begin{pmatrix} Q_{\Gamma_1,\mathcal{F}} & 0 \\ 0 & Q_{\Gamma_2,\mathcal{F}} \end{pmatrix}
$$

where $Q_{\Gamma_1,\mathcal{F}} = diag(q_l)$, $q_l = 1$ if $1 \leq l \leq q$ and $q_l = 0$ if $q < l$. Then we write the preconditioner

$$
M_{ARAS(q)}^{-1} = (I + R_{\Gamma}^T D(I_n - Q_{\mathcal{F}}\hat{P})^{-1} - I_n)CR_{\Gamma})M_{RAS}^{-1} \tag{4.37}
$$

As previously we introduce a matrix $\bar{Q}_{\mathcal{F}} = (I - Q_{\mathcal{F}})$.

We can then follow the demonstration done in the proof of theorem 4.3 writing the error on the interface in the Fourier space as

$$
\hat{u}_{|\Gamma}^\infty - \hat{u}_{|\Gamma}^k = \hat{u}_{|\Gamma}^\infty - \bar{Q}_{\mathcal{F}}\hat{P}\hat{u}_{|\Gamma}^{k-1} - Q_{\mathcal{F}}\hat{u}_{|\Gamma}^\infty - \bar{Q}_{\mathcal{F}}\hat{c} \tag{4.38}
$$

Thus,

$$
\hat{e}_{|\Gamma}^k = \bar{Q}_{\mathcal{F}}\left(\hat{u}_{|\Gamma}^\infty - \hat{P}\hat{u}_{|\Gamma}^{k-1} - \hat{c}\right) \tag{4.39}
$$

Regarding equation (4.15), $\hat{P}\hat{u}_{|\Gamma}^{k-1} + \hat{c} = \hat{u}_{|\Gamma}^k$, and then,

$$
\hat{e}_{|\Gamma}^k = \bar{Q}_{\mathcal{F}}\hat{e}_{|\Gamma}^k \tag{4.40}
$$

As $\hat{e}_{|\Gamma}^k = \hat{P}\hat{e}_{|\Gamma}^{k-1}$ we write

$$
\hat{e}_{|\Gamma}^k = \bar{Q}_{\mathcal{F}}\hat{P}\hat{e}_{|\Gamma}^{k-1} \tag{4.41}
$$

We showed that the ARAS iterative process has an error transfer operator, $\bar{Q}_{\mathcal{F}}\hat{P}$, equal to the part of the error transfer operator $\hat{P}$ that we did not compute. We note that $||\bar{Q}_\lambda\hat{P}|| \leq ||\hat{P}|| < 1$.

We can apply Proposition 4.2 and write:

$$
\rho(T_{ARAS(q)}) = \rho(\bar{Q}_{\mathcal{F}}\hat{P}) \tag{4.42}
$$

The conclusion becomes the same as applying Theorem 4.3.

We pointed out here the way the approximation of the error's transfer operator affects the convergence of Schwarz iterative processes in the case of a separable operator for a two domain decomposition. This enables us to understand the philosophy of approximating the matrix P in different spaces and links the works done in [24, 2, 26].

5. Results on academic problems.

5.1. $2D$ theoretical study. The goal of this section is to validate the ARAS method on a simple case where the ARAS preconditioner can be written analytically and where we can apply Theorem 4.3. We consider the $2D$ problem decomposed in 2 domains presented in subsection 4.4. The grid size is about 32×32. This subsection provides the theoretical framework we implement in Matlab. Here, we verify numerically the theoretical results given previously.

We build the matrix $\hat{P} \in \mathbb{C}^{30 \times 30}$. Only the internal points are taken, leading to 30 modes which can be accelerated. Those modes decrease from 0.8106 to 0.1531. We decide to compute the entire $\hat{P}$ and a truncated one of size $q = 15$, $Q_{\mathcal{F}}\hat{P}$. Figure 5.1 shows the coefficient of the matrix computed. The goal here is to retrieve the convergence rate given by the application of theorem 4.3 in equation (4.33). The convergence rate of a ARAS(q) type preconditioner is related to the coefficients of $\hat{P}$ denoted by δ_l.
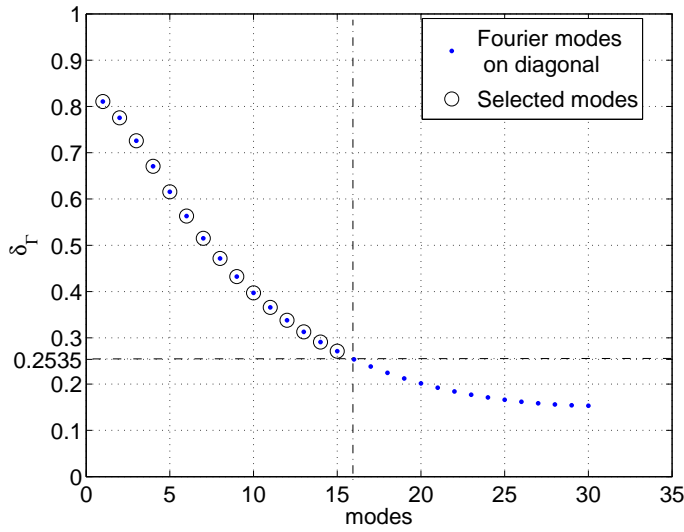

FIGURE 5.1. *Diagonal coefficients of $\hat{P}$ and $Q_{\mathcal{F}}\hat{P}$ corresponding to the modes to accelerate.*

For $q = 15$ we note that,

$$\delta_1 = 0.8106 \tag{5.1}$$

$$\delta_{q+1} = 0.2535 \tag{5.2}$$

$$\delta_{q+1}^2 = 0.0643 \tag{5.3}$$

Then we compute the RAS, ARAS(q) and ARAS2(q) preconditioners and compute for each preconditioner of type $*$ the spectral radius of T_* and the conditioning number in the 2 norm of $M_*^{-1}A$. The convergence for each appropriate stopping criteria is 10^{-10}.

Table 5.1 shows the numerical convergence rate of RAS, ARAS(15), ARAS2(15), and ARAS2(30) for the test case presented in Subsection 4.4. The numerical values obtained for ρT_* match perfectly the theoretical results. It exhibits also that the Aitken acceleration of the RAS enhances the condition number of the preconditioned problem. In accordance with the theory, when $q = 30$, the size of the artificial interface, P is exact and $M_{ARAS2}^{-1} = A^{-1}$ numerically.

prec. $*$	$\rho(T_*)$	$\kappa(M_*^{-1}A)$	It. Rich.	It. GCR
RAS	0.8106	30.0083	96	18
ARAS(q=15)	0.2535	5.2358	14	7
ARAS2(q=15)	0.0643	1.1451	7	5
ARAS2(q=30)	1.4319 e-13	1.0000	1	1

TABLE 5.1

Numerical performance of RAS, ARAS and ARAS2 on the 2D Poisson problem.

Moreover the number of iterations of the iterative process ARAS(q=15) is twice the number of iterations of the iterative process ARAS2(q=15).

5.2. Observing the influence of the partitioning and the approximation space on a 2D Helmholtz problem. Here, we focus on the influence of the partitioning chosen to set up the domain decomposition method. We also focus on the influence of the choice of a base to approximate the Aitken's acceleration. When the mesh is known it is possible to partition the operator following a geometric partitioning. One point is to see what can happen if we partition the operator with a graph partitioning approach such as METIS. Another point is to see how the choice of a base influence the performance and the cost of the ARAS type preconditioner.

Let us consider the 2D Helmholtz problem $(-\omega - \triangle)u = f$ in $\Omega = [0,1]^2$, $u = 0$ on $\partial\Omega$. The problem is discretized by second order finite differences with m points in each direction x and y giving a space step $h = \frac{1}{m-1}$. The set value $\omega = 0.98\frac{4}{h^2}(1 - cos(\pi h))$ is close to the minimum eigenvalue of the discrete $-\triangle$ operator in order to have an ill-conditioned discrete problem with $\kappa_\infty(A) = 1.7918\,E + 07$ for $m = 164$.
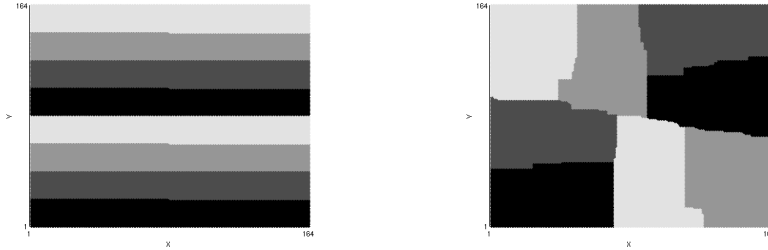

FIGURE 5.2. *Partitioning into 8 parts on a 2D Helmholtz problem of size 164×164, (left) Physical Band partitioning, (right) METIS partitioning.*

First we solve the problem with a physical band partitioning, and then we solve it with a METIS partitioning using eight sub-domains. Figure 5.2 illustrates the physical band and the METIS partitioning using eight sub-domains. In the physical partitioning, borders are smooth, contrary to the METIS partitioning which creates corners and irregular borders. The corners give cross points which deteriorate the convergence of the Schwarz method.

For each partitioning, we build the ARAS2 preconditioner in two different bases:

- an orthogonal base arising from the application of the preconditioner RAS on a sequence of random vectors (see subsection 3.1).
- a base built from the successive Schwarz solution on the interface and passed in its SVD base (see subsection 3.2).

Remark 7. In the following, we denote by ARAS2($r=\frac{n}{q}$) the preconditioner approximated in the "random" base. Because the number of vectors can be high for this kind of base, we choose to express the reduction number r in parentheses instead of q, the number of column of $\mathbb{U}_q$, but the formula is still:

$$M^{-1}_{ARAS(r=\frac{n}{q}),\delta} = \left(I_m + R_\Gamma^T \mathbb{U}_q \left(\left(I_q - \tilde{P}_{\mathbb{U}_q}\right)^{-1} - I_q\right)\mathbb{U}_q^{-1} R_\Gamma\right)\sum_{i=1}^{p}\tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta}$$

Figure 5.3 (respectively Figure 5.4) presents the Richardson process with ARAS2 and the ARAS2 preconditioned GCR Krylov method for the physical band partitioning (respectively a METIS partitioning) using a random base or a SVD base. These results were obtained with a sequential Matlab code able to run small academic problems. The Krylov method used is the gradient conjugate residual GCR while the LU factorisation is used to solve the sub-domain's problems.

These results exhibit:

- Richardson processes in Figure 5.3 converge while in Figure 5.4 only the RAS iterative process converges and the ARAS2 process diverges. Consequently the physical partitioning enables good convergence of the RAS which can be accelerated with the approximation of P by $P_{\mathbb{U}_q}$. The METIS partitioning slows the convergence of the RAS due to the cross points.Then, using a domain decomposition method as a solver with an algebraic partitioning can produce bad results when it is accelerated by Aitken. Let us notice that the full P makes the RAS process converge in one iteration.
- Nevertheless, the Aitken-RAS used as a preconditioner is very efficient even on the METIS partitioning with cross points where the Aitken-RAS as a Richardson process diverges. This makes the Aitken-RAS a robust algebraic preconditioner. We must notice that the effect of the preconditioning with a METIS partitioning is less efficient than the one with the physical partitioning.
- The better the base $\mathbb{U}_q$ is able to represent the interface solution, the better the preconditioner is for the random base and the SVD base.

Let us observe the difference between the two choices of base to compute the acceleration. On the one hand, the choice of an orthogonal base arising from the application of the RAS preconditioner on a sequence of random vectors presents good advantages for a preconditioner. With this approach, the preconditioner can be used for different right-hand sides. However, the number of vectors necessary to describe the interface can be close to the size of the global interface, increasing the cost of the preconditioner. On the other hand, it is possible to build the acceleration for many iterations of the Additive Schwarz process, computing the SVD of the interface solutions. Then the acceleration process is problem-dependent, but experience shows that a small number of iterations can enable a good approximation of $P_{\mathbb{U}_q}$.

For a physical partitioning, we can evaluate the cost of each preconditioner. For each sub-domain, the artificial interface is of size 164 for the uppermost or lowermost sub-domain, and $164 * 2 = 328$ for internal sub-domains. Hence, for $p = 8$ the global interface has a size of $164 * (6 * 2 + 2) = 2296$. For $r = 1$ the base is complete and $P_{\mathbb{U}_q}$ is exact. There is no need to use ARAS as a preconditioning technique. For all r the size of Γ is $\frac{2296}{r}$. Then the number of $M_{RAS}^{-1}x = y$ products to build M_{ARAS}^{-1} is $3 * \frac{2296}{r}$. While the number of $M_{RAS}^{-1}x = y$ products to build M_{ARAS}^{-1} for the base arising from SVD only depends on the number of Richardson iterations.

Figure 5.3, shows that for $r = 8$, $n = 185$ and the number of products $M_{RAS}^{-1}x = y$ is 555, the convergence of GCR is reached in 11 iterations. For 24 iterations of Schwarz, we build a
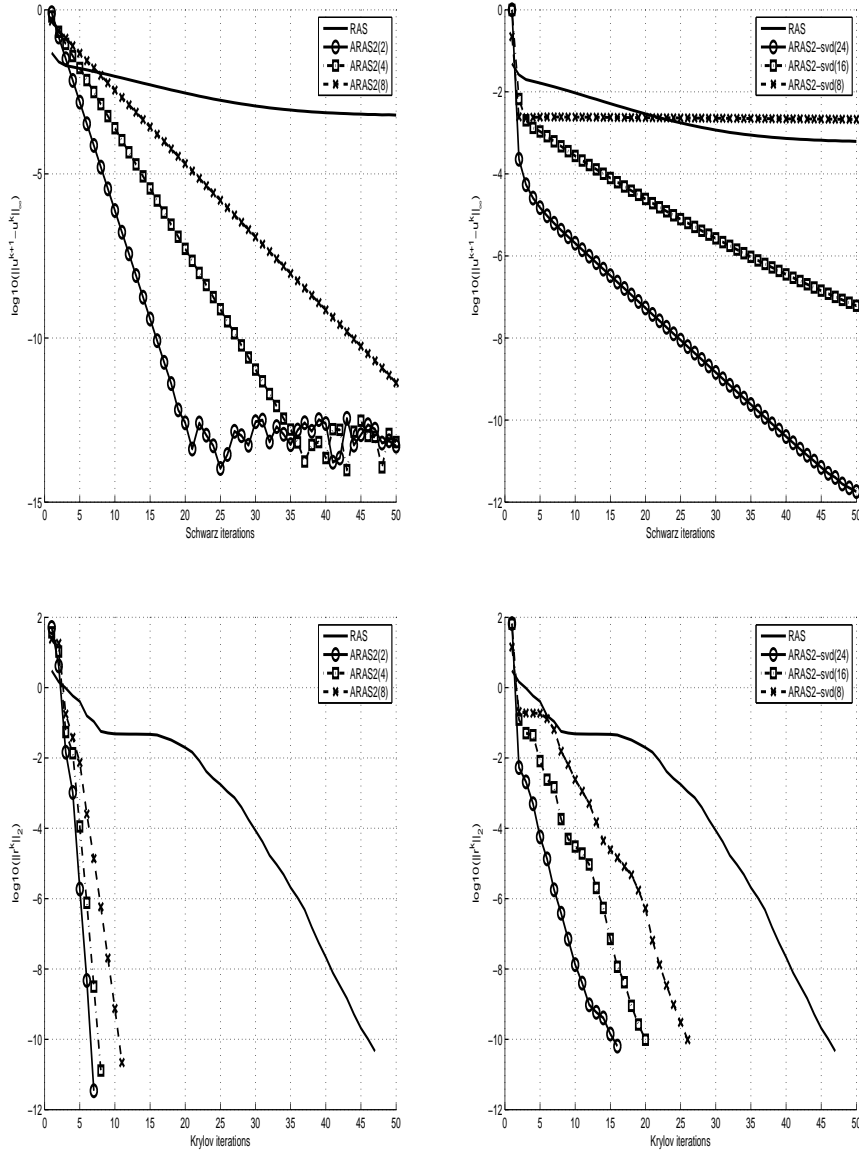
FIGURE 5.3. *Solving 2D Helmholtz equation on a* 164×164 *Cartesian grid, physical band partitioning,* $p = 8$,*(left) ARAS2($r = \frac{n}{q}$) is built with a Random base, (right) ARAS2(q) is built with a SVD base, (top) Convergence of Iterative Schwarz Process, (bottom) convergence of GCR method preconditioned by RAS and ARAS2.*

matrix $P_{\mathbb{U}_q}$ of size 24, which is around eight times smaller than with the previous base. The number of matrix products is 48, 12 times smaller than with the previous base. The number of GCR iterations is 15. Eventually, the cost of a good independent preconditioner is excessive compared to the one with the SVD.

Figure 5.5 focuses on the eigenvalue of the error transfer operators when the base is computed from SVD and both partitioning. We compute all the singular values corresponding
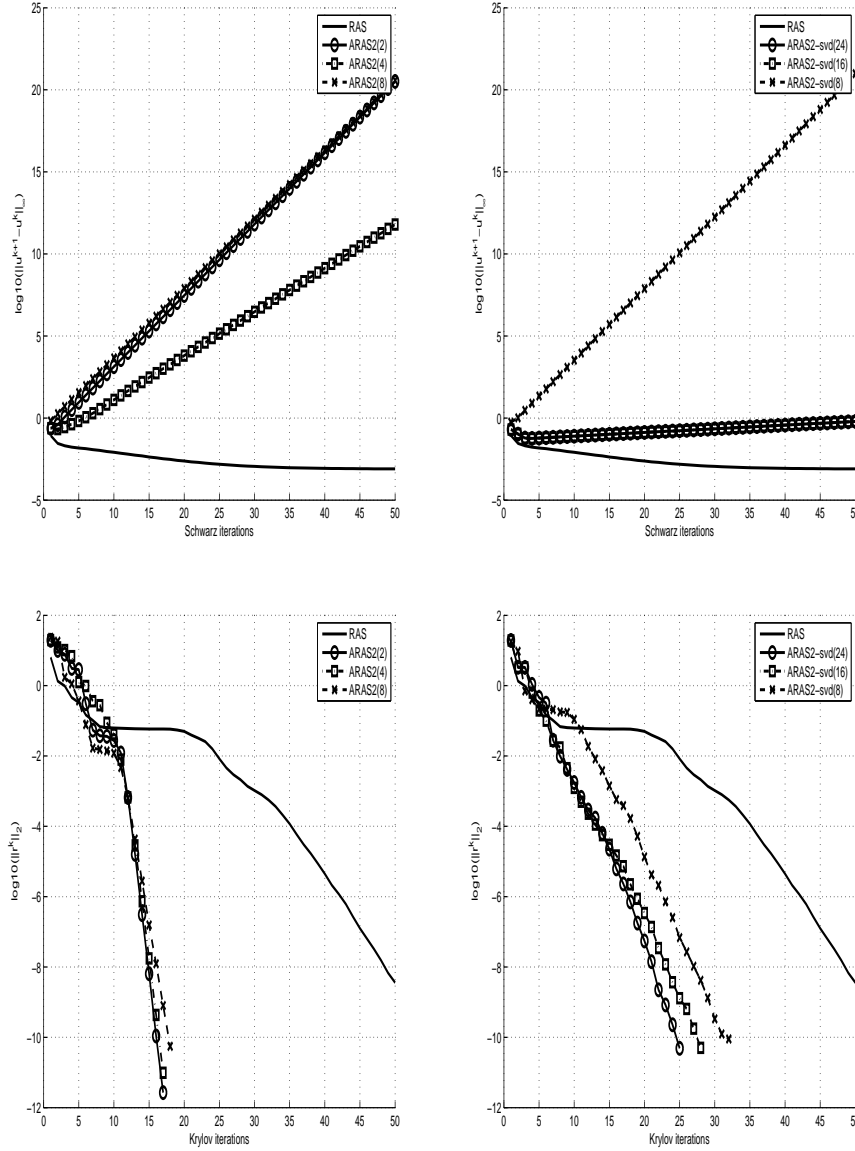
FIGURE 5.4. *Solving 2D Helmholtz equation on a* 164×164 *Cartesian grid, METIS partitioning,* $p = 8$,*(left) ARAS2 is built with a Random base, (right) ARAS2 is built with a SVD base, (top) Convergence of Iterative Schwarz Process, (bottom) convergence of GCR method preconditioned by RAS and ARAS2.*

to the number of interface points and select 24 singular values from this set of n values. For 8 partitions with a manual partitioning, $n = 2296$ and with a METIS partitioning we obtain 1295 interfaces points. We saw that the Aitken-RAS technique used as a Richardson iterative process diverges for the METIS partitioning. We study the spectrum of a preconditioner in the two cases and compare it to the spectrum of the RAS preconditioning method. For convenience we consider only a set of the 40 largest eigenvalues.

The predicted values of the spectrum of the error transfer operator on the interface gives
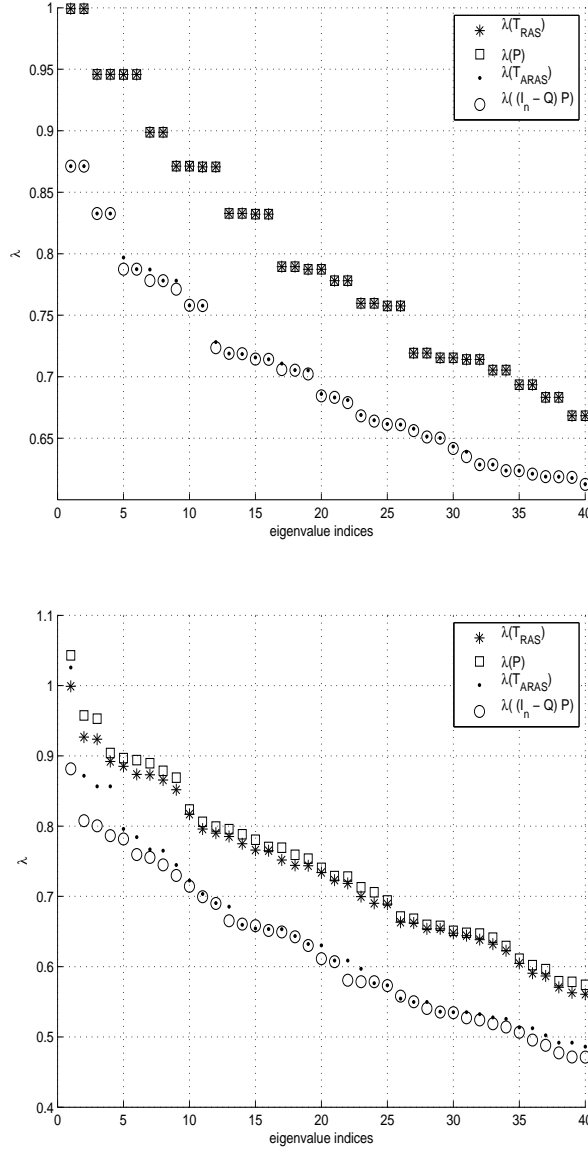
FIGURE 5.5. *Eigenvalues of* $T_{ARAS}(q = 24)$ *compared to eigenvalues of* P_{U_q} *for a* 164×164 *Cartesian grid,* $p = 8$,*(top) Band partitioning, ARAS is built with a SVD base computed with* 24 *singular vectors chosen over* 2296, *(bottom) METIS partitioning, ARAS is built with a SVD base computed with* 24 *singular vectors chosen over* 1295.

a good approximation of the spectrum of the iterative process in the case of the band partitioning. Otherwise, for the METIS partitioning, it appears that the spectrum gives a good idea of the spectrum of the iterative method but differs for the first eigenvalues. Then it appears that the first eigenvalue of T_{ARAS} is greater than 1 instead of the value computed for $\bar{Q}P$, which is less than 1. It should explain why the iterative process diverges.

This empirical analysis shows the influence of a partitioning technique on the Schwarz

preconditioner. It is important to know that if the user has the entire knowledge of the linear system he solves then he should provide a physical partitioning which can have smooth boundaries. Otherwise, if the sub-problems are non-singular then the Schwarz method used as a preconditioning technique is efficient but can present a lack of speed in the convergence. The second point is the choice of a good base. The two bases are efficient, but the one arising from the SVD presents the best choice for time computing considerations.

6. Results on industrial problems. In section 5 we pointed out that a graph partitioning to define the domain decomposition and the algorithm of approximation using the SVD of the Schwarz solutions should be a good choice to algebraically build the ARAS preconditioner. We first propose an estimation of the computing cost and then apply the preconditioner to an industrial problem.

6.1. Computing cost modelling. We want to evaluate the cost of building and applying an ARAS type preconditioner in terms of arithmetic complexity. We denote by $\mathcal{AC}(*)$ the arithmetic complexity of an operation $*$. Let considers a matrix $A \in \mathbb{R}^{m \times m}$. This matrix is decomposed into p sub-matrices $A_i \in \mathbb{R}^{m_i \times m_i}$. The decomposition leads to have an interface Γ of size n. The coarse interface is of size q. We denote by $x_\alpha \in \mathbb{R}^\alpha$ where $\alpha \in \mathbb{N}$ should be any of m, m_i, n or q.

6.1.1. Arithmetic complexity of applying an ARAS type preconditioner. Let considers the operation:

$$M_{RAS}^{-1} x_m = \sum_{i=1}^{p} \tilde{R}_i^T A_i^{-1} x_{m_i} = y$$

The cost of such an operation mostly consists in the p operations $A_i^{-1} x_{m_i}$ which depends on the cost of a chosen method to inverse A_i such as a Krylov methods or a LU factorization or maybe an incomplete LU factorization.

Then the cost of applying a RAS preconditioner is written as:

$$\mathcal{AC}\left(M_{RAS}^{-1} x_m\right) = p \times \mathcal{AC}\left(A_i^{-1} x_{m_i}\right) \tag{6.1}$$

We now consider the operation:

$$M_{ARAS(q)}^{-1} x_m = \left(I_m + R_\Gamma^T \mathbb{U}_q \left(\left(I_q - P_{\mathbb{U}_q}\right)^{-1} - I_q\right) \mathbb{U}_q^T R_\Gamma\right) \sum_{i=1}^{p} \tilde{R}_i^T A_i^{-1} R_i x_m$$

The cost of such an operation consists in one application of a RAS preconditioner, the base transfer operated by $\mathbb{U}_{\shortparallel}$ and solving $\left(I_q - P_{\mathbb{U}_q}\right) y_q = x_q$. Others or summing operations: 1 sum between 2 vectors of size n and, one subtract between two vector of size q.

Then the cost of applying a RAS preconditioner is written as:

$$\mathcal{AC}\left(M_{ARAS}^{-1} x_m\right) = p \times \mathcal{AC}\left(A_i^{-1} x_{m_i}\right) + \mathcal{AC}\left(\mathbb{U}_q^T x_m\right)$$
$$+ \mathcal{AC}\left(\left(I_q - P_{\mathbb{U}_q}\right)^{-1} x_q\right) + \mathcal{AC}\left(\mathbb{U}_q x_q\right)$$
$$+ \mathcal{AC}\left(x_q - x_q\right) + \mathcal{AC}\left(x_n + x_n\right)$$

We note that:
- An addition between 2 vectors of size n consists in n operations.
- A subtraction between 2 vectors of size n consists in n operations.

- A scalar product between 2 vectors of size n consists in n product and n sum.
- A multiplication between a matrix with n lines and m columns and a vector with m lines consists in n scalar products of vectors of size m.

Then we write,

$$\mathcal{AC}\left(x_n + x_n\right) = n$$
$$\mathcal{AC}\left(x_q - x_q\right) = q$$
$$\mathcal{AC}\left(\mathbb{U}_q x_q\right) = m \times 2 \times q$$
$$\mathcal{AC}\left(\mathbb{U}_q^T x_m\right) = q \times 2 \times m$$

And,

$$\mathcal{AC}\left(M_{ARAS}^{-1} x_m\right) = p \times \mathcal{AC}\left(A_i^{-1} x_{m_i}\right) \tag{6.2}$$

$$+ \mathcal{AC}\left(\left(I_q - P_{\mathbb{U}_q}\right)^{-1} x_q\right) \tag{6.3}$$

$$+ 4 \times q \times m + n + q \tag{6.4}$$

Remark 8. In most cases the coarsening is such that $q \ll m$. Then the cost

$$\mathcal{AC}\left(\left(I_q - P_{\mathbb{U}_q}\right)^{-1} x_q\right)$$

should be very small compared to the cost $p \times \mathcal{AC}\left(A_i^{-1} x_{m_i}\right)$. If it is the case

$$\mathcal{AC}\left(M_{ARAS(q)}^{-1} x_m\right) = \mathcal{AC}\left(M_{RAS}^{-1} x_m\right) + \mathcal{O}(m) \tag{6.5}$$

This means that the cost of one application of the ARAS preconditioner is close to the cost of one application of the RAS preconditioner when $q \ll m$.

We now estimate the cost of applying an ARAS2 preconditioner:

$$M_{ARAS2(q)}^{-1} x_m = 2 M_{ARAS(q)}^{-1} - M_{ARAS(q)}^{-1} A M_{ARAS(q)}^{-1} x_m \tag{6.6}$$

It consists in 2 applications of ARAS(q) and one matrix vector product on the entire domain. We note that the matrix A is sparse. So, denoting by $nnz(A)$ the number of non zeros elements of A we write,

$$\mathcal{AC}(A x_m) = 2 \times nnz(A)$$

Hence,

$$\mathcal{AC}\left(M_{ARAS2(q)}^{-1} x_m\right) = 2 \times \mathcal{AC}\left(M_{ARAS(q)}^{-1} x_m\right) + 2 \times nnz(A) + 2\mathcal{O}(m) \tag{6.7}$$

Remark 9. For $q \ll m$,

$$\mathcal{AC}\left(M_{ARAS2(q)}^{-1} x_m\right) = 2 \times \mathcal{AC}\left(M_{RAS}^{-1} x_m\right) + \mathcal{O}(nnz(A)) + \mathcal{O}(m)$$

6.1.2. Arithmetic complexity of building the coarse space $\mathbb{U}_q$ **and** $P_{\mathbb{U}_q}$. We focus here on the cost to build a base arising from the SVD of the Schwarz solutions on the interface. We refer to Algorithm 5 which proposes a robust way to implement the Aitken's acceleration without inversion.

We compute $q + 2$ iterations of a RAS iterative process. It consists in applying the preconditioner on a vector x_m and summing the result with another vector of the same size. We can write

$$\mathcal{AC}(T_{RAS}x_m) = \mathcal{AC}(M_{RAS}^{-1}x_m) + \mathcal{O}(m) \tag{6.8}$$

Then we perform a SVD over a set of $q + 2$ vectors of size n, $X_{q+2} \in \mathbb{R}^{n \times (q+2)}$.

$$\mathcal{AC}(\text{building } \mathbb{U}_q) \leq (q+2) \times \mathcal{AC}(T_{RAS}x_m) + \mathcal{AC}(SVD(X_{q+2})) \tag{6.9}$$

After this, we apply one iteration of the Schwarz iterative process on at most the q first left singular vectors to build $P_{\mathbb{U}_q}$. Thus,

$$\mathcal{AC}(\text{building } \mathbb{U}_q \text{ and } P_{\mathbb{U}_q}) \leq (q+2) \times \mathcal{AC}(T_{RAS}x_m)$$
$$+ \mathcal{AC}(SVD(X_{q+2})) + q \times \mathcal{AC}(M_{RAS}^{-1}x_m) \tag{6.10}$$

Hence,

$$\mathcal{AC}(\text{building } \mathbb{U}_q \text{ and } P_{\mathbb{U}_q}) \leq 2(q+1) \times \mathcal{AC}(M_{RAS}^{-1}x_m) + \mathcal{AC}(SVD(X_{q+2})) + \mathcal{O}(m) \tag{6.11}$$

The cost of building the coarse space and the error transfer operator depends on the number q of vectors needed. Then the ARAS(q) preconditioner will be a good choice compared to RAS if q is sufficiently small compared to the number of application of the preconditioner involved in the Krylov iterative method.

Remark 10. This computation, following the robust algorithm 5, is nearly two times costly than Algorithm 4 with inversion. In fact, the building of $P_{\mathbb{U}_q}$ with Algorithm 4 consists in inverting an error matrix of size q and multiplying it, on the left, by another matrix of size q. For simplicity we consider those operations of order $\mathcal{O}(m)$.

$$\mathcal{AC}(\text{building } \mathbb{U}_q \text{ and } P_{\mathbb{U}_q}) \leq (q+2) \times \mathcal{AC}(M_{RAS}^{-1}x_m) + \mathcal{AC}(SVD(X_{q+2})) + \mathcal{O}(m) \tag{6.12}$$

In order to save computing, Algorithm 4 with inversion is the best choice.

6.1.3. Parallelization. It is important to note that the Restricted Additive Schwarz process is fully parallel, in the sense that the inverse of A_i can be computed independently by every single process i handling a domain i. Then for p processes, we can re-write the formula (6.1):

$$\mathcal{AC}\left(M_{RAS}^{-1}x_m\right) = \mathcal{AC}\left(A_i^{-1}x_{m_i}\right) \tag{6.13}$$

The parallelism leads to have a reduction of the matrix vector product. Then we write, for p processes:

$$\mathcal{AC}\left(\mathbb{U}_q x_q\right) = m_i \times 2 \times q$$
$$\mathcal{AC}\left(\mathbb{U}_q^T x_m\right) = q \times 2 \times m_i$$
$$\mathcal{AC}(Ax_m) = 2 \times nnz_i(A)$$

Table 6.1 shows the parallel arithmetic complexity in $\mathcal{O}(m_i)$ for building and applying a parallel ARAS(q) preconditioner.

Task	$\mathcal{AC}(SVD(X_{q+2}))$	$\mathcal{AC}\left(A_i^{-1}x_{m_i}\right)$
Building	1	$q+2$
Apply	0	1

TABLE 6.1

Parallel complexity for building and applying an ARAS(q) preconditioner

6.2. Application on a $3D$ CFD industrial case. We consider CASE_017 RM07 available in the sparse matrix collection [13], which represents a $3D$ viscous case with a "frozen" turbulence. Here, the geometry is a jet engine compressor. The problem is discretized among 54527 nodes. Seven variables per node are considered. The resulting matrix is of size 381689 with 37464962 non-zeros. The matrix is not symmetric.

We use a PETSc-MPI implementation with a PARMETIS partitioning taking into account the block structure and the weight of each block. The matrix is partitioned in four parts, with a minimum overlap. We apply the ARAS and ARAS2 left-preconditioners with 40 basis vectors computed following Algorithm 5.

Figure 6.1 shows the convergence of the preconditioned GMRES and the convergence of the Richardson processes associated to the preconditioners. In order to disminish the cost of building phase, the tolerance is set to 10^{-6} for the local solution, but only for the building phase.Then, the iterations to build $P_{\mathbb{U}_q}$ take less cpu-time and memory allocation than one application of the preconditioner during the solution phase. The ARAS(q) preconditioner has been presented as a left preconditioner. The stopping criteria used for the GMRES method implemented in PETSc is based on the relative residual. This is why the curves are not going to the same tolerance. We discuss this point with the following results. For a minimal overlap, without knowledge of the underlying equations and mesh design, the ARAS preconditioner is efficient, and also its mutliplicative version, ARAS2.

Table 6.2 shows the performance results corresponding to Figure 6.1 on an SGI Altix Xe340. While applying ARAS(40) or ARAS2(40), both Solution Time and Memory allocation are reduced. The time involved in the building phase of the preconditioner depends of the choice of local factorization and the kind of algorithm chosen. The different times follows the formula of the arithmetic complexity presented in Subsection 6.1. The relative residual for RAS and ARAS(q) is the same. We explain the difference of convergence between ARAS and ARAS2 for this case by the fact that the Richardson processes diverge and leads to an amplified inaccurary in ARAS2.

Prec.	Building Time (s.)	Solution Time (s.)	Max. Loc. Mem (M.O.)	$\|Ax-b\|_2/\|b\|_2$
RAS	8.684	1552.89	1068	1.5704 e-09
ARAS (40)	429.943	1086.63	1048	9.7492e-10
ARAS2 (40)	446.454	1174.97	1010	2.2760e-07

TABLE 6.2

Solving 3D Navier Stokes equation (CASE RM07), PARMETIS partitioning with weights, $p = 4$, overlap 0, ARAS2 is built with a SVD basis, GMRES method preconditioned by RAS and ARAS (built with tol = 10^{-6}).

7. Conclusion. We presented an integration of the Aitken acceleration technique in the RAS preconditioning. This integration leads to a multi-level preconditioner. One level is the entire domain, while the second is the entire artificial interface. Since the computation of the error transfer operator is costly, we propose an algebraic computation of a coarse space, built
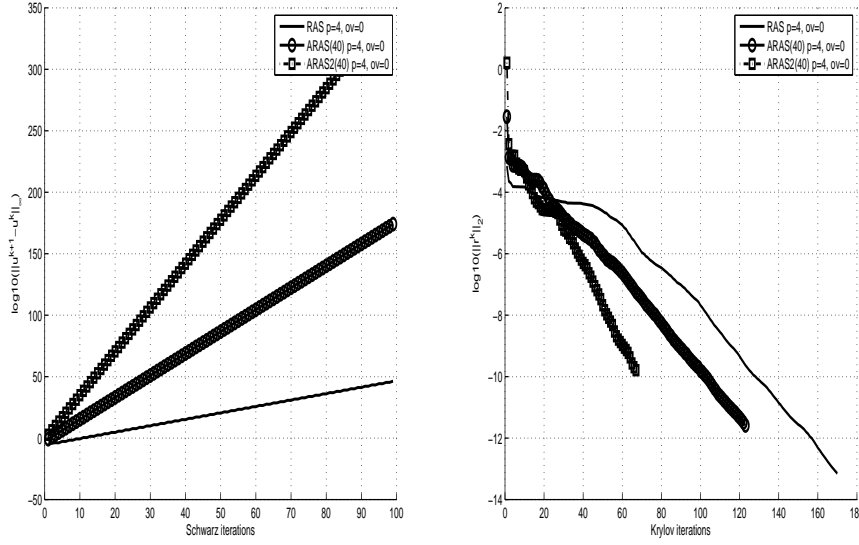
FIGURE 6.1. *Solving 3D Navier Stokes equation (CASE RM07), PARMETIS partitioning with weights, $p = 4$, overlap 0, ARAS2 is built with a SVD basis, (left) Convergence of Iterative Schwarz Process, (right) convergence of GMRES method preconditioned by RAS and ARAS (built with $tol = 10^{-6}$).*

from the SVD decomposition of Schwarz solutions on the interface. The results is a cheap and fully algebraic enhancement of the RAS preconditioner. An analysis of the convergence of this preconditioner is given when the basis is built analytically and shows the effect of the preconditioner depending on the choice of the mode to be accelerated. Finally, a result is provided on a $3D$ industrial case without knowledge of the underlying equations and the mesh design.

Future work should focus on other algorithms to build algebraically the Aitken acceleration in order to reduce the time spent in the building time. Another issue concerns work on partitioning techniques for domain decomposition. Actually, there is no technique to provide local system with insurance of inversion.

REFERENCES

[1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, 1996, ch. 5.
[2] J. BARANGER, M. GARBEY, AND F. OUDIN-DARDUN, *The aitken-like acceleration of the schwarz method on nonuniform cartesian grids*, SIAM J. Sci. Comput., 30 (2008), pp. 2566–2586.
[3] N. BARBEROU, M. GARBEY, M. HESS, M. RESH, T. ROSSI, J. TOIVANEN, AND D. TROMEUR-DERVOUT, *Efficient metacomputing of elliptic linear and non-linear problems*, Journal of Parallel and Distributed Computing, 63 (2003), pp. 564–577.

[4] R. BEAUWENS, *Iterative solution methods*, Appl. Numer. Math., 51 (2004), pp. 437–450.

[5] J. H. BRAMBLE, J. E. PASCIAK, AND A. H. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring. I*, Math. Comp., 47 (1986), pp. 103–134.

[6] X.-C. CAI, M. DRYJA, AND M. SARKIS, *A restricted additive Schwarz preconditioner with harmonic overlap for symmetric positive definite linear systems*, Cubo, 6 (2004), pp. 73–95.

[7] X.-C. CAI, W. D. GROPP, AND D. E. KEYES, *A comparison of some domain decomposition and ILU preconditioned iterative methods for nonsymmetric elliptic problems*, Numer. Linear Algebra Appl., 1 (1994), pp. 477–504.

[8] X.-C. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797 (electronic).

[9] L. M. CARVALHO, L. GIRAUD, AND P. LE TALLEC, *Algebraic two-level preconditioners for the Schur complement method*, SIAM J. Sci. Comput., 22 (2000), pp. 1987–2005.

[10] L. M. CARVALHO, L. GIRAUD, AND G. MEURANT, *Local preconditioners for two-level non-overlapping domain decomposition methods*, Numer. Linear Algebra Appl., 8 (2001), pp. 207–227.

[11] P. CHEVALIER AND F. NATAF, *An optimized order 2 (OO2) method for the Helmholtz equation*, C. R. Acad. Sci. Paris Sér. I Math., 326 (1998), pp. 769–774.

[12] P. G. CIARLET, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, 1994, ch. 5.

[13] T.A. DAVIS AND Y. HU, *The university of florida sparse matrix collection*. ACM Transactions on Mathematical Software (to appear), 20YY. http://www.cise.ufl.edu/research/sparse/matrices.

[14] M. DRYJA AND O. B. WIDLUND, *Some domain decomposition algorithms for elliptic problems*, in Iterative methods for large linear systems (Austin, TX, 1988), Academic Press, Boston, MA, 1990, pp. 273–291.

[15] T. DUFAUD AND D. TROMEUR-DERVOUT, *Adaptive aitken-schwarz method for non separable operator on multiprocessor systems*, in Parallel Computational Fluid Dynamics Recent Advances & Future Directions, Rupak Biswas and NASA Advanced Supercomputing Division NASA Ames Research Center, eds., DEStech Publications, 2010, pp. 297–305.

[16] ———, *Aitken's acceleration of the restricted additive Schwarz preconditioning using coarse approximations on the interface*, C. R. Math. Acad. Sci. Paris, 348 (2010), pp. 821–824.

[17] E. EFSTATHIOU AND M. J. GANDER, *Why restricted additive Schwarz converges faster than additive Schwarz*, BIT, 43 (2003), pp. 945–959.

[18] S. C. EISENSTAT, H. C. ELMAN, AND M. H. SCHULTZ, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[19] B. ENGQUIST AND H.-K. ZHAO, *Absorbing boundary conditions for domain decomposition*, Appl. Numer. Math., 27 (1998), pp. 341–365. Absorbing boundary conditions.

[20] B. FLANNERY, W. H. PRESS, S. TEUKOLSKY, AND W. VETTERLING, eds., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, third ed., 2007.

[21] A. FRULLONE AND D. TROMEUR-DERVOUT, *A new formulation of nudft applied to aitken-schwarz ddm on nonuniform meshes*, in Parallel Computational Fluid Dynamics 2005, 2006, pp. 493–500.

[22] M. J. GANDER, L. HALPERN, F. MAGOULÈS, AND F.-X. ROUX, *Analysis of patch substructuring methods*, Int. J. Appl. Math. Comput. Sci., 17 (2007), pp. 395–402.

[23] M. J. GANDER, F. MAGOULÈS, AND F. NATAF, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 24 (2002), pp. 38–60 (electronic).

[24] M. GARBEY, *Acceleration of the Schwarz Method for Elliptic Problems*, SIAM J. Sci. Comput., 26 (2005), pp. 1871–1893.

[25] M. GARBEY AND D. TROMEUR-DERVOUT, *Two level domain decomposition for multiclusters*, in 12th Int. Conf. on Domain Decomposition Methods DD12, T. Chan & Al editors, ed., ddm.org, 2001, pp. 325–339.

[26] ———, *On some aitken like acceleration of the schwarz method*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 1493–1513. LMS Workshop on Domain Decomposition Methods in Fluid Mechanics (London, 2001).

[27] L. GERARDO-GIORDA AND F. NATAF, *Optimized Schwarz methods for unsymmetric layered problems with strongly discontinuous and anisotropic coefficients*, J. Numer. Math., 13 (2005), pp. 265–294.

[28] W. D. GROPP AND D. E. KEYES, *Parallel performance of domain-decomposed preconditioned Krylov methods for PDEs with locally uniform refinement*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 128–145.

[29] B. N. KHOROMSKIJ AND G. WITTUM, *Numerical solution of elliptic differential equations by reduction to the interface*, vol. 36 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2004.

[30] Z. LI AND Y. SAAD, *SchurRAS: a restricted version of the overlapping Schur complement preconditioner*, SIAM J. Sci. Comput., 27 (2006), pp. 1787–1801 (electronic).

[31] P.-L. LIONS, *On the Schwarz alternating method. I*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987), SIAM, Philadelphia, PA, 1988, pp. 1–42.

[32] ———, *On the Schwarz alternating method. III. A variant for nonoverlapping subdomains*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX,

1989), SIAM, Philadelphia, PA, 1990, pp. 202–223.

[33] L. D. MARINI AND A. QUARTERONI, *A relaxation procedure for domain decomposition methods using finite elements*, Numer. Math., 55 (1989), pp. 575–598.

[34] RAMESH NATARAJAN, *Domain decomposition using spectral expansions of Steklov-Poincaré operators. II. A matrix formulation*, SIAM J. Sci. Comput., 18 (1997), pp. 1187–1199.

[35] A. QUARTERONI AND A. VALLI, *Domain decomposition methods for partial differential equations*, Numerical Mathematics and Scientific Computation, The Clarendon Press Oxford University Press, New York, 1999. Oxford Science Publications.

[36] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive Schwarz preconditioning*, SIAM J. Sci. Comput., 29 (2007), pp. 2402–2425 (electronic).

[37] O. STEINBACH, *A natural domain decomposition method with non-matching grids*, Appl. Numer. Math., 54 (2005), pp. 362–377.

[38] J. STOER AND R. BULIRSCH, *Introduction to numerical analysis*, vol. 12 of Texts in Applied Mathematics, Springer-Verlag, New York, third ed., 2002. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.

[39] D. TROMEUR-DERVOUT, *Meshfree adaptative aitken-schwarz domain decomposition with application to darcy flow*, in PARALLEL, DISTRIBUTED AND GRID COMPUTING FOR ENGINEERING, BHV Topping and P Ivanyi, eds., vol. 21 of Computational Science Engineering and Technology Series, SAXE-COBURG PUBLICATIONS, 2009, pp. 217–250.

[40] D. M. YOUNG AND K. C. JEA, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.

33